Digital Business Ecosystem

Contract n° 507953

Workpackage 6

Self Organisation

Deliverable 6.1

Self-Organisation in Multi-Agent Systems

Project funded by the European Community under the "Information Society Technology" Programme

Contract Number: 507953
Project Acronym: DBE
Title: Digital Business Ecosystem

Deliverable N°: D6.1
Due date: 31/07/04
Delivery Date: 31/07/04

Short Description:
A definition for self-organisation, based upon information theory and entropy, is presented for DBE services in the Evolutionary Environment, where services are considered as software agents.

The results of the simulations created to test the measure are presented, and consideration is given to the problem of 'linkage disequilibrium'. The measure of self-organisation is quantitative, so the work presented here also partially addresses the following deliverable, 'D6.2 The control of self-organisation, and a performance measure'.

Partners owning: ICL
Partners contributed: ICL
Made available to: Consortium and EC

Versioning		
Version	**Date**	**Author, Organisation**
1	31/07/2004	Gerard Briscoe, ICL

Quality check
1st Internal Reviewer: Paolo Dini
2nd Internal Reviewer: Thomas Heistracher
3rd Internal Reviewer: Maria Petrou

Entropy-Based Complexity Measure for the Evolution-Based Self-Organisation of Agent Populations

G. Briscoe
Imperial College London

July 2004

Abstract

The specific objective is to provide a performance measure for the organisation of the DBE services(modelled as agents) into applications(service-chains) in the DBE.

A definition for self-organisation, based on statistical physics, is studied for evolving DBE service populations. The measure calculates the entropy in the population to determine the randomness, which is then used in determining the organisational complexity. An efficiency measure is then presented for the organisational complexity present, relative to the maximum possible in the population, which is useful for detecting clustering.

Keywords: self-organisation, agents, complexity, entropy, evolution, population, clustering

Acronyms

Below is an extensive list of the acronyms used within this document.

BML	=	Business Modelling Language
DBE	=	Digital Business Ecosystem
DNA	=	deoxyribose nucleic acid (genetic information)
DOP	=	DBE Open Protocol
ebXML	=	e-business eXtensible Markup Language
EC	=	Evolutionary Computing
EvE	=	Evolutionary Environment
ESS	=	Evolutionary Stable Strategies
FL	=	Fitness Landscape
KC	=	Kolmogorov-Chaitin
MAS	=	Multi-Agent System
MDL	=	Minimum Description Length
MFT	=	Mean Field Theory
PDA	=	Personal Digital Assistant
SDL	=	Service Description Language
SM	=	Service Manifest
SME	=	Small & Medium (sized) Enterprises
SOAP	=	Simple Object Access Protocol
SOC	=	Self-Organised Criticality
STU	=	Salzburg Technical University
TM	=	Turing Machine
UTM	=	Universal Turing Machine
VLP	=	variable length population
WSDL	=	Web Services Definition Language

Contents

1 Introduction

A definition for measuring organisation is sought for the composition of DBE Service Manifests(SMs) within the Population Objects of the Evolutionary Environment(EvE)[10], where Evolutionary Computing(EC) is used to find the optimal SM-chain to a user request. To minimise the dependency on specific terms in favour of general concepts, an abstract model based on a Multi-Agent System(MAS) will be used to represent the EvE, specifically focusing on the evolution of SM-chains.

Self-organisation is an emergent global 'behaviour' of complex systems, which consist of many autonomous components with local interaction rules. The DBE SMs are the components of the EvE system which interact with one another in an evolving population of SM-chains to provide, at the global level, optimal solutions to a user request. The scope of this document is concerned with evolutionary self-organisation of a single population in the EvE.

In the following section, Section 2, an abstract model of the DBE service composition will be created and justified, along with any other necessary assumptions.

Section 3 is a summary of the literature review performed, including the philosophical meaning of organisation and of self in organisation, and the applicable measures of organisation will be described and considered. In Section 4, the measure chosen to be investigated further will be introduced, and then adapted for evolving agent populations. Its suitability as a measure of organisation in evolving agent populations will then be considered.

The hypotheses to be tested will be described with the simulation scenarios for their testing, in Section 5. The results from these simulations will then be shown and analysed, in Section 6. Concluding remarks regarding the achievements and implications will be made in Section 7, with the future work discussed in Section 8.

Appendix A is an **Interdisciplinary Dictionary** of the key terms used within this document to facilitate communication within the consortium. Appendix **??** defines the numerous **acronyms** used, and Appendix B shows the spreadsheet tool created to allow experiments with the organisation measure defined in Section 4. Appendix D is the presentation slides presented at the recent STU Mozart meeting, which were used to introduce this work.

2 Assumptions

2.1 Modelling DBE Services as Software Agents

DBE services consist of an executable component(interfaced by SDL) and a descriptive component(BML data). DBE services are often considered as next-generation web services, and their initial implementation technologies are considered to be extensions to existing web services technologies. Examples include the consideration of extending SOAP(Simple Object Access Protocol) to define the DOP(DBE Open Protocol), extending ebXML(eBusiness eXtensible Markup Language) for the BML(Business Modelling Language), and extending WSDL(Web Services Definition Language) for the SDL(Service Description Language)[17].

The World Wide Web is used more and more for application to application communication. The programmatic interfaces made available are referred to as web services[41]. So, web services can be considered as loosely coupled, reusable software components that semantically encapsulate discrete functionality[38].

A software agent can be defined as follows: A computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its designed objectives[42].

The concept that next-generation web services will be software agents is a current topic of research[23]. There is also much work in combining web services and agents to create hybrid systems, in the same way in which PDAs and mobile phones are merging. Examples include agent systems which use web service technologies[27], or web services which use agent ontology-based information processing to determine their actions[20].

Considering the information presented, it is reasonable to model DBE services as software agents, and therefore the DBE service ecosystem (EvE) as a Multi-Agent System(MAS) with Evolutionary Computing(EC).

2.2 Evolutionary Computing

The populations of DBE services undergoing EC in the DBE service ecosystem(EvE) will be modelled as populations of agents undergoing EC in the MAS model. These populations will exist to search the agent combinatorial space to find solutions to SME user requests. The fitness of individuals within a population will be determined by a fitness function, which will be based primarily on comparing the descriptive components (BML data) of SM-chains with the user requests (BML data). The EC populations are found in the Evolutionary Environment(EvE)[10], formerly found in the Fitness Landscape(FL)[13].

2.3 BML (Business Modelling Language)

The Business Modelling Language exists to allow business people to represent the business processes of their respective companies. These business processes include the services that they offer, and the parameters associated with these services[12].

2.4 Multi-Agent System Model of DBE Evolutionary Environment

This abstract Multi-Agent System model is of the Evolutionary Environment[10]. DBE services are agents, and therefore SM-chains are agent-chains. Although the model supports multiple Habitats from the EvE, the scope of this document is with one habitat, and the organisation of a single Population Object within that habitat. The habitat is necessary to provide the pool of agents(DBE services) from which solutions are evolved.

This multi-agent system model is a mobile agent system, where agents move to different agent stations (nodes of the MAS), this is akin to the service migration in the Evolutionary Environment. At these agent stations, Evolutionary Computing(EC) is used to compose the fittest(optimal) agent-chains.

The evolving software agents each consist of a remotely referenced executable component(SDL interface) and an ontology-based description(BML). The description contained within each agent acts as a guarantee of its functionality, and is the inheritable component from one generation to the next. Although there is no constraint on the language used for the executable components, they should have compatible interfacing(SDL), so that agents have the ability to aggregate into chains to perform more complex processes. The set of agents available at an agent-station is used as a gene pool for evolving a population of agent-chains. The evolution is directed by a user request (selection pressure), which consists of a multi-process description (BML data).

As both the agents and the user request are functionally described in the same language (BML), the fitness will be based on comparing the descriptive components(BML data) of the agents with the complex description(BML data) of the user request (selection pressure). The agent-chains(SM-chains) are then evolved over many generations to determine the optimal solution. Mutations can occur by switching components in and out of the chain structure. Recombination (Crossover) may occur by combining elements of two or more agent-chains into a new agent-chain.

3 Literature Review

A review of the available literature on self-organisation will be presented, with respect to its general properties, its application to MASs, and its application to populations in MASs. Self-organisation has been around since the late 1940s[6], but has escaped general formalisation. There have been many attempts at creating a general formalisation of self-organisation[29, 14]. This will not be the case here, because it shall be argued that any definition of self-organisation is context dependent, in the same way that the choice of statistical measures is dependent on the data being analysed.

3.1 Philosophy of Organisation

The philosophy of organisation is complicated, because organisation has different meanings to different people. There have been many notions and definitions of organisation, useful within their different contexts. They have come from cybernetics, thermodynamics, mathematics, information theory, synergetics, and others. Many people use the term self-organising, but it has no generally accepted meaning, as the abundance of definitions suggests.[19]

Proposing a definition or organisation faces the cybernetic problem of defining system, the cognitive problem of perspective, the philosophical problem of defining self, and the universal problem of defining organisation.[19]

3.1.1 System

The system in this context is the evolutionary system, which includes a population of agent-chains, recombination of the agent-chains, replication of the agent-chains from one generation to the next, and a selection pressure which causes differential fitness between the agent-chains.

3.1.2 Perspective

This can be defined as the perception of the observer, in perceiving the organisation of a system. This matches the intuitive definition, Ill know it when I see it[36]. Although this intuitive definition may make formalisation difficult, it does show that organisation is perspective-dependent, i.e. relative to the context in which it occurs.

In the context of an evolutionary system, the observer does not exist in the traditional sense, but is the selection pressure(environment). Therefore, for an evolutionary system, the organisation of its population is relative to its environment.

3.1.3 Self

Whether a system is self-organising or merely organised depends on whether or not the process causing the organisation is an internal component of the system under consideration. Although somewhat simplistic, it does intuitively make sense and relegates the argument to defining the boundaries of the system being considered, in order to

determine if the force causing organisation is internal or external to the system. For an evolving population the force leading to organisation within the population is the selection pressure, which is formed by the environment of the populations existence and the competition between the individuals of the population. As these are internal components of the system, the system is self-organising. Put simply, an evolutionary system without a selection pressure is not evolving, and therefore would not be an evolving system.

3.1.4 Organisation

Now that definitions for the system for which organisation is context dependent, the perspective to which it is relative, and the self by which it is caused have been proposed, a definition of organisation can be considered. To visualise the context, an evolving population of agent-chains which lack a 2D or 3D metric space, it is necessary to consider it in a more abstract form. We will let a single square ☐ represent an agent, with different colours to represent the different agents. Agent-chains will be represented by a sequence of squares ☐☐☐. A population is represented by multiple agent-chains, as shown below:

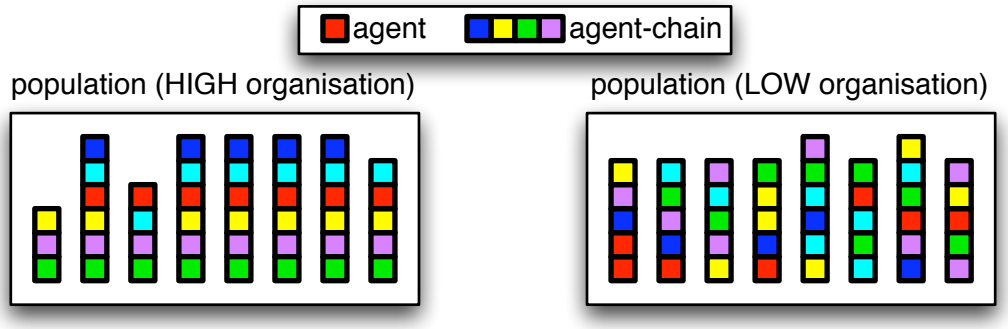


Figure 1: Visualisation of Organisation in Agent-chain Populations

For Figure 1, the number of agents in total and of each type(colour) is the same in both populations. However, the population of agent-chains on the left intuitively shows organisation through the uniformity of the colours across the agent-chains (clustering), whereas the population to the right shows little or no organisation. The organisation of a system is the clustering of its components, in this case the agents, into coherent patterns and structures, where the complexity of these patterns and structures can be a measure of its **organisational complexity**. So the organisational complexity measures the information content in a population of agent-chains.

3.2 Current Self-Organisation Definitions

There are biologically inspired definitions of self-organisation defined for MASs without evolution[30], such measures, although valuable in their own right, are not applicable to evolving agent populations. Many alternative techniques have been proposed to model self-organisation within populations in general, and agent populations specifically. Each with their own definition of what property or properties demonstrate organisation of, or within, a population. The possible applicable alternatives will be considered for their

suitability in defining the organisational complexity (organisation) of a population of agent-chains.

The relationship of **fitness** and self-organisation is made confusing by the multiple definitions of both. The widely accepted definition of fitness is that it is a property of individual phenotypes, and is a measure of the ability to produce mature offspring in the next generation which themselves will be able to reproduce. The self-organisation sought is a property of an entire population, not an individual. Therefore, fitness cannot be the self-organisation of a population. The two are indeed connected and will be discussed more, once a definition of self-organisation is determined.

Self-organised criticality(SOC) in evolution is defined as punctuated equilibrium, in which the populations critical state is when the fitness of the individuals is uniform, and an avalanche is caused by the appearance and spread of advantageous mutations within the population, which temporarily disrupts the uniformity of individual fitness across the population. Whether this process displays SOC remains unclear. There are those who claim that SOC is demonstrated by the available fossil data[39] with a power law distribution on the lifetimes of genera drawn from fossil records, and artificial life simulations[1] with a power law distribution on the lifetimes of competing species. On the other hand, there are those who feel that the fossil data is inconclusive, and the artificial life simulations do not show SOC, because the key power law behaviour in both can be generated by models without SOC[28]. SOC has nothing to say about the organisation of the population at the critical state, only the organisation of the events, avalanches, which moves the population temporarily away, and then back to the critical state.

One possibility for determining the organisational complexity (organisation) of the population of agent-chains would be the application of the **Minimum Description Length**(MDL) principle[7] to the executable components of the agent-chains in the population. The best model, among a collection of tentatively suggested ones, is the one that gives the smallest stochastic complexity to the given data. However, the MDL principle has nothing to say about how to select the family of model classes to be applied for determining the stochastic complexity. In fact, this problem cannot be adequately formalised[34]. In practice, their selection is based on human judgement and prior knowledge of the kinds of models that have been used in the past. This limitation would make it impossible to automate the organisational complexity estimation process, due to the necessity of human intervention at model selection for every different user request.

Mean Field Theory (MFT) requires a neighbour model to describe the interactions between neighbours in the systems it is applied to, and is therefore easily applied to Cellular Automata[21]. The main concept in MFT is that, for a single particle, the most important contribution to its interactions comes from its neighbouring particles, and therefore its behaviour can be approximated by relying upon the mean field caused by its neighbouring particles. MFT application to the evolution of a population[18] requires a neighbour model, which in actual biological systems is a reasonable assumption. Agent populations lack neighbour models based on a 2D or 3D metric space. The only available neighbour model becomes a distance measure on a parameter space measuring dissimilarity. However, such a neighbourhood model cannot represent the information-based interactions between individuals of a population of agent-chains.

In **Replicator dynamics**, of evolutionary game theory, agents of a population play a game and do not optimise over strategic alternatives, but inherit a fixed strategy and then replicate depending on the strategys payoff (fitness)[16]. An equilibrium, a stable steady state, can be reached in which all the strategies have the same expected payoff. It is called a stable steady state, because if pushed slightly off, it returns back to the equilibrium of the stable steady state. An evolutionarily stable strategy (ESS) is an equilibrium strategy that can overcome the presence of a small number of invaders, so it is an asymptotically stable steady state, which is a more stringent stable equilibrium concept[40], than a stable steady state. The self-organisation found in replicator dynamics is not the composition of the population directly, but the presence of stable steady states, in which the genotype frequencies of the population cease to change from one generation to the next. The composition of the resulting population can even be randomly distributed, but stable. This self-organisation measure is more relevant to the genetic stability of the population from one generation to the next, rather than the organisational complexity (organisation).

Kolmogorov-Chaitin(KC) complexity measures the complexity of binary sequences by the smallest possible Universal Turing Machine (UTM), algorithm (program and input), that produces the sequence. A sequence is said to be regular if the algorithm necessary to produce it on a UTM is shorter than the sequence itself. A regular sequence is said to be compressible, whereas its compression into the most succinct UTM is said to be incompressible, as it cannot be reduced any further in length. A purely random sequence is said to be incompressible, because the UTM to represent it cannot be shorter than the random sequence itself. This intuitively makes sense from the point of view of algorithmic complexity, because algorithmically regular sequences require a shorter program to produce them. To measure a population of sequences, the KC-complexity would be the shortest UTM to produce the entire population of sequences. Chaitin himself has considered the application of KC-complexity to evolution[11], and realised that although KC-complexity represents a satisfactory definition of randomness in algorithmic information theory, it is not so useful in biology. For example applying it to physical structures, one sees that a gas is the most random, and a crystal the least random, but neither has any significant biological organisation. For populations of evolving agent-chains, this problem manifests itself most significantly when the agents are randomly distributed within the agent-chains of the population, having maximum KC-complexity, rather than a complexity of zero which it ought to have. This property makes KC-complexity unsuitable as a measure of the organisational complexity (organisation) of a population of agent-chains.

The **Prugell-Bennett Shapiro formalism** models the evolutionary dynamics of a population of sequences, using statistical mechanics techniques, focusing on replica symmetry[31]. The individual sequences are not considered directly, but in terms of the statistical properties of the population. So, it is described as using a macroscopic level of description, and the particular statistical properties, which are used to characterise the population, are called macroscopics. A macroscopic formulation of an evolving population reduces the huge number of degrees of freedom to the dynamics of a few quantities. A non-linear system of a few degrees of freedom can be readily solved or numerically iterated. So, the formalism can predict the optimal amount of selection. However, since a macroscopic description throws information away, human insight is essential so that the appropriate macroscopics are chosen[37]. There is no procedural method for determining

the macroscopics, which would make it impossible to automate, due to the necessity of human intervention at every different user request.

A measure called the **Physical complexity** can be estimated for a population of sequences, calculated from the difference between the maximal entropy of the population, and the actual entropy of the population when in its environment[4]. **In effect, the environment(selection pressure) entropically weights the combination space of sequences, which is shown by the composition of the population.** This Physical complexity is based on Shannons entropy measure of information, and measures the amount of information contained in the population about its environment, and is therefore conditional on its environment. The Physical complexity can be estimated by counting the number of loci per sequence that are fixed in the population. The measure is however formulated for a population of sequences with the same length, and can be inaccurate if there is significant epistasis, which is when the loci are interdependent upon one another. So the estimates from counting the loci become inaccurate, which in-turn negatively effects any estimate of complexity[5].

3.3 Summary

Any definition chosen should define the organisational complexity (organisation), the clustering, of the agents within an evolving population of agent-chains, with no initial constraints on the number of genotypes or phenotypes, or models specifically seeking speciation. Neither will organisation be defined by the presence or lack of evolutionary stable strategies or self-organised criticality. The organisational complexity sought after is the clustering within a population in general, without the specific inclusion of the models previously mentioned, but capable of representing their appearance in the population.

None of the proposed measures are directly applicable as a measure of organisational complexity (organisation) for a population of agent-chains. Self-organised criticality is concerned with the organisation of events affecting the population, rather than the organisation within the population. Mean Field Theory is not applicable due to its necessity of a neighbour model for defining interaction. Replicator dynamics measures the genetic stability of the population, rather than the organisational complexity. KC-complexity is not applicable as random sequences, and randomness in general, have maximum complexity. The Prugell-Bennett Shapiro formalism also cannot be automated, as it requires human intervention to choose the appropriate properties of the population, for constructing the model.

The Physical complexity for a population of sequences is in essence the organisational complexity (organisation), because it estimates complexity based upon the individuals of the population within the context of their environment. The formulation is not without problems, as it is for fixed length populations, can suffer from epistatic effects, and the mapping of loci to agents needs to be considered. However, none of the problems are fundamental properties of the measure, so they should not be insurmountable.

The use of Physical complexity as a measure of the organisational complexity will be investigated further to determine its suitability.

4 Measure of Organisation

The Physical complexity measure is currently not applicable to the population of agent-chains envisaged in the target system. It needs to be modified, redefining the assumptions for the different conditions in which it will be used. First, the measure will be introduced in more detail.

4.1 Existing Physical Complexity

In this subsection the motivation and mathematics of Physical Complexity will be given, before it is adapted and modified.

4.1.1 Origins of Physical Complexity

Physical complexity was born from the need to determine the proportion of information in sequences of DNA. It has long been established that the length of DNA sequences is an unsatisfactory measure of the information content, as they contain significant redundancy. So the information contained is not directly proportional to the length[5]. Understanding the DNA requires knowing the environment(context) in which it exists.

The purpose of DNA in the context of life may initially sound straightforward, as DNA is considered to be language of life. However, consider viruses which can consist of only a strand of DNA covered by a protein coating. Scientists are still split on whether viruses are alive or not. The only function viruses perform is to replicate, not showing any of the other commonly accepted signs of life in more complex organisms, such as respiration, growth, etc. We shall consider viruses to be alive, and accept that their only function is to replicate. The process of replication requires resources, energy and matter, to be harvested. Viruses are the simplest form of life known. More complex forms of life have evolved far more complex, specialised, specific and effective ways to acquire the necessary energy and matter for replication.

So consider that for any individual the environment represents the problem of extracting energy for replication. Then the DNA sequence of an individual represents a solution to this problem. This hopefully alleviates the misconception that the DNA of individuals encodes their environment; it does not, it encodes a solution to extract energy and matter from their environment for its replication. Furthermore, the individual DNA solution is not a simple inverse of the 'problem' that the environment represents.

Even with this understanding, the problem remains of the need to define the environment, to be able to distinguish the information from the redundancy in the solution. This situation is resolved in the Physical complexity measure by analysing a group of solutions to the same problem. The consistency between the different solutions shows the information, and the differences are the redundancy. Entropy is a measure of disorder, as it measures the number of states accessible to a system with equal probability. A large number of accessible states is usually associated with disorder. So entropy is used to determine the redundancy from the information, in a population of solutions. The measure

therefore provides a context-relative measure of organisational complexity, by measuring the population, which relives us of the need to define the context(environment).

4.1.2 Definition of Physical Complexity

Physical complexity is derived[4] from the notion of conditional complexity defined by Kolmogorov, which is different from traditional Kolmogorov complexity.. It states that the determination of complexity of a sequence is conditional on the environment in which the sequence is interpreted. The traditional Kolmogorov complexity, however, is only conditional on the implicit rules of mathematics, and nothing else. These rules are necessary to interpret the program on the tape of the Turing Machine(TM), which is conceptually correct, but requires more refinement. Consider a TM that takes a tape e as input (which represents its physical environment) and that includes the particular rules of mathematics of this world. Without such a tape, this TM is incapable of computing anything, except for writing to the output what it reads in the input. Thus, in the absence of tape e all sequences s have maximal KC-complexity, because there is nothing by which to determine regularity.

The conditional complexity can be stated as $K(s|e)$[24], as the length of the smallest program that computes sequence s from environment e, where $C_T(p, e)$ denotes the result of running program p on Turing machine T given input sequence e.

$$K(s|e) = \min \{|p| : s = C_T(p, e)\} \tag{1}$$

This is not yet the Physical complexity. Rather, it is the smallest program that computes sequence s from environment e, in the limit of sequences of infinite length, and only contain bits that are entirely unrelated to e, since, if they were not, they could be obtained from e with a program of size tending to zero. The physical complexity $K(s : e)$ can now be defined as the number of bits that are meaningful in sequence s (that can be obtained from e with a program of vanishing size), and is given by the mutual complexity[24].

$$K(s : e) = K(s|\emptyset) - K(s|e) \tag{2}$$

$K(s|\emptyset)$ is the unconditional complexity, i.e. the complexity given an empty input tape $e \equiv \emptyset$. This is different from the Kolmogorov complexity because, in Kolmogorovs construction, the rules of mathematics were given to the TM. As argued above, every sequence s is random if no environment e is specified, as non-randomness can only exist with respect to a specific world, or environment. Thus, $K(s|\emptyset)$ is always maximal, given by the length of s:

$$K(s|\emptyset) = |s| \tag{3}$$

So (2) represents the length of the sequence s minus those bits that cannot be obtained from e. So, conversely (2) represents the number of bits that can be obtained from

a sequence s, by a computation with vanishing program size, from e. Thus $K(s:e)$ represents the Physical complexity of s.

The determination of the Physical complexity $K(s:e)$ of sequence s given a description of the environment e., is not practical, meaning that it cannot, in general, be determined by inspection.

In other words, it is impossible to determine which, and how many, of the bits of sequence s correspond to information about the environment e. The reason is that, in general, we are unaware of the coding used to code information about e in s, and as a consequence coding and non-coding bits look entirely alike. However, it is possible to distinguish coding from non-coding bits if we are given multiple copies of a symbolic sequence that have adapted independently to the environment within which it is to be interpreted, or more generally, if a statistical ensemble(population) of sequences is available to us. In that case, coding bits are revealed by nonuniform probability distributions across the population (conserved sites), whereas random bits sport uniform distributions (volatile sites). The determination of complexity then becomes an exercise in information theory. The average complexity $\langle K \rangle$, in the limit of infinitely long sequences tends to the entropy of the sequences s in the population(ensemble) S:

$$\begin{aligned} \langle K(s) \rangle_S &= \sum_{s \in S} p(s) K(s) \approx H(S) \\ &= -\sum_{s \in S} p(s) \log p(s) \end{aligned} \tag{4}$$

Entropy H is defined as follows. At a basic level, Boltzmanns entropy is a (logarithmic) measure of the number of accessible states (with equal probability) to a system:

$$S = k_B log(g) \tag{5}$$

where k_B is Boltzmanns constant, with units of energy/kelvin, and g is the multiplicity function. Shannons entropy is also a logarithmic measure. At its simplest, conceptual level, for a binary alphabet, it is given by the simple function

$$H = log_2(M) \tag{6}$$

where M is the number of symbols. This is none other than the number of bits needed to encode a set of M binary numbers. For $M = 512$, $H = 9$, etc[35]. If each symbol is equally probable, we can rewrite the above function as

$$\begin{aligned} H &= -log_2(1/M) \\ &= -log_2(p) \end{aligned} \tag{7}$$

where p is the probability of occurrence of any one of the symbols. For a source that outputs an infinite sequence of bits to communicate a finite set of symbols M, Shannon generalised the above function to express an average symbol length. The derivation is easier to see for a large but finite total number of symbols N. N_i is the number of occurrences of the symbol M_i, and therefore acts as a weight in the average:

$$\begin{aligned} H &= -log_2(1/M) \\ &= -log_2(p) \end{aligned} \tag{8}$$

$$\begin{aligned} H &= \frac{\sum_{i=1}^{M} N_i \left[-\log(1/M_i)\right]}{\sum_{i=1}^{M} N_i} = \frac{\sum_{i=1}^{M} N_i \left[-\log(1/M_i)\right]}{N} \\ &= -\sum_{i=1}^{M} \frac{N_i}{N} \left[\log(1/M_i)\right] = -\sum_{i=1}^{M} p_i \log(p_i) \end{aligned} \tag{9}$$

So Shannons formula for entropy is an average logarithmic measure of the symbol sets, i.e. the average word size. In general it does not have to apply to binary systems, so the logarithm can be taken to the appropriate base.

(4) remains consistent with (3) as the determination of $K(s|\emptyset)$, string s in the absence of an environment e, must equal the sequences length $|s|$. Indeed, if nothing is known about the environment that sequence s pertains to, the probability distribution $p(s)$ must be uniformly random according to the principle of insufficient reason. Johann Bernoullis principle states that if we are ignorant of the ways an event can occur (and therefore have no reason to believe that one way will occur preferentially compared to another), the event will occur equally likely in any way. As a consequence, the maximum entropy of a population(ensemble) is equivalent to the length of the sequences, which is the cardinality of the sequences, in the population(ensemble) $H(S) = |s|$. On the other hand, if an environment e is given we have some information about the system, and the probability distribution is nonuniform. Indeed, it can be shown that for every probability distribution $p(s|e)$ to find sequence s given environment e, we have as a result of the concavity of Shannon entropy:

$$H(S|e) \leq H(S|\emptyset) = |s| \tag{10}$$

The difference between the maximal entropy $H(S|\emptyset) = |s|$ and $[H(S|e)$, according to the construction outlined above, should then represent the average number of bits in sequence s taken from the population(ensemble) S that can be obtained by zero-length universal programs from the environment e.

$$\begin{aligned} \langle K(s:e) \rangle_S &= \sum_{s \in S} p(s) K(s:e) \approx H(S|\emptyset) - H(S|e) \\ &\equiv I(S|e) \end{aligned} \tag{11}$$

So, $I(S|e)$ is the information about the environment e stored in the population S, which we identify with the Physical complexity. To estimate $I(S|e)$, it is necessary to estimate the entropy $H(S|e)$, using a representative population(ensemble) S (given environment e) of such sequences, by the sum of the probability $p(s|e)$, of a sequence s in the population S given the environment e, multiplied by the logarithm of the probability $p(s|e)$:

$$H(S|e) = -\sum_{s \in S} p(s|e) \log p(s|e) \tag{12}$$

The entropy $H(S|e)$ can be estimated by summing the per-site $H(i)$ entropies of the sequence, where i is a site in the sequence s.

$$H(S|e) \approx \sum_{i=1}^{|s|} H(i) \tag{13}$$

Random sites are identified by a nearly uniform probability distribution, and contribute positively to the entropy, whereas non-random sites (which have strongly peaked distributions) contribute very little. So the physical complexity of a sequence s in a population(ensemble) S is $\langle K(s:e) \rangle_S$, which will be abbreviated to C, and is the maximal entropy $H(S|\emptyset)$ minus the sum of the per-site entropies.

$$\langle K(s:e) \rangle_S = C = H(S|\emptyset) - \sum_{i=1}^{|s|} H(i) \tag{14}$$

For clarity, the length of the sequences s, which is the cardinality of the sequences $|s|$, will be abbreviated to ℓ:

$$|s| \equiv \ell \tag{15}$$

Sequences s of length ℓ are constructed from an alphabet, a set D. The size of the alphabet being the cardinality of the set D. The probability $p_d(i)$, is the probability that a site i, in the sequences s, takes on character d from the alphabet D, rather than any other character. So the sum of the $p_d(i)$ probabilities equals one. So, the per site entropy $H(i)$, for each site i, of the sequences s, is the sum over the alphabet D, and is defined as:

$$\begin{gathered} H(i) = -\sum_{d \in D} p_d(i) \log_{|D|} p_d(i) \\ where: \ 1 \leq i \leq \ell, 0 \leq p_d(i) \leq 1, \sum_{d \in D} p_d(i) = 1 \\ given: \ i = site, \ell = length, D = alphabet, |D| = alphabetsize \end{gathered} \tag{16}$$

Taking the log to the base $|D|$ results in $H(i)$ ranging between 0 and 1.

$$0 \leq H(i:e) \leq 1 \tag{17}$$

If a site i is identical across the population, it will have no entropy.

$$H_{\min}(i) = 0 \tag{18}$$

A site has maximum entropy if the probabilities $p_d(i)$ in (16) are equal. In effect the content of the site is uniformly random. This is true for all i:

$$\begin{gathered} H_{\max}(i) = 1 \\ given: \ p_d(i) = \tfrac{1}{|D|} \end{gathered} \tag{19}$$

When the entropy is minimum, i.e. zero, then the site i holds information, as every sample shows the same character of the alphabet. When the entropy is at its maximum, the character found in the site i is uniformly random and therefore site i holds no information. Therefore the amount of information is the maximal entropy of the site (20), minus the actual per-site entropy (16):

$$\begin{aligned} I(i) &= H_{\max}(i) - H(i) \\ &= 1 - H(i) \end{aligned} \tag{20}$$

DNA, whose sequence encodes the genetic information of living organisms, was the original driver for the creation of this measure. So it is a good example upon which to demonstrate the measure. DNA sequences are made up from four nucleotides, Adenosine, Thymine, Cytosine and Guanine.

Figure 2: Visualisation of Organisation in Agent-chain Populations

The nucleotides always pair as follows, Adenosine with Thymine, and Cytosine with Guanine. So DNA sequences are reduced to a genome, just half of their paired information:

TGCGATACCTTTTGATTGG

Figure 3: Genome of DNA Sequence

Given a sufficiently sized sample population, the $p_d(i)$ probabilities can be estimated by the frequencies of nucleotides at the sites. Consider the samples of the genome below.

site	1	2	3	4	5	6	7	8	9	10	**11**	12	13	14	15	**16**	17	**18**	19
Sample 1:	C	G	C	G	A	T	A	C	C	T	T	T	T	G	A	T	T	G	G
Sample 2:	C	G	C	G	A	T	A	C	C	T	A	T	T	G	A	T	T	G	G
Sample 3:	C	G	C	G	A	T	A	C	C	T	G	T	T	G	A	T	T	C	G
Sample 4:	C	G	C	G	A	T	A	C	C	T	C	T	T	G	A	T	T	C	G

Figure 4: Samples of the same Genome

If the per-site entropy for site 11, in Figure 4, is calculated, it will have maximum entropy, as the nucleotides(characters of the alphabet) all have equal probability.

$$
\begin{aligned}
given: \quad & D = alphabet = \{A, T, C, G\}, |D| = alphabet\ size = 4 \\
& p_A(11) = p_T(11) = p_C(11) = p_T(11) = \frac{1}{4} \\
hence: H(11) &= -\sum_{d \in D}^{A,T,C,G} p_d(11) \log_{|D|} p_d(11) \\
&= -\left(\frac{1}{4}\log_4\frac{1}{4} + \frac{1}{4}\log_4\frac{1}{4} + \frac{1}{4}\log_4\frac{1}{4} + \frac{1}{4}\log_4\frac{1}{4}\right) \\
&= -\left(-\frac{1}{4} + -\frac{1}{4} + -\frac{1}{4} + -\frac{1}{4}\right) = 1
\end{aligned}
$$

As the per-site entropy(randomness) is maximum, the information content is minimum, i.e. zero.

$$
\begin{aligned}
I(11) &= 1 - H(11) \\
&= 1 - 1 = 0
\end{aligned}
$$

This intuitively makes sense, and states that if the site content is random across the population, then there is no information.

If the per-site entropy for site 16, in Figure 4, is calculated, it will have no entropy, as one nucleotide, Thymine, has a probability of one, and the remaining three have probabilities of zero.

$$
\begin{aligned}
given: \quad & D = alphabet = \{A,T,C,G\} |D| = alphabetsize = 4 \\
& p_A(16) = p_T(16) = p_C(16) = p_T(16) = 1 \\
hence: \; H(16) &= -\sum_{d \in D}^{A,T,C,G} p_d(16) \log_{|D|} p_d(16) \\
&= -\left(0 \log_4 0 + 1 \log_4 1 + 0 \log_4 0 + 0 \log_4 0\right) \\
&= -\left(0 + 0 + 0 + 0\right) = 0
\end{aligned}
$$

As the per-site entropy is minimum, the information content is maximum.

$$
\begin{aligned}
I(16) &= 1 - H(16) \\
&= 1 - 0 = 1
\end{aligned}
$$

This also intuitively makes sense, as it states that if the site is identical across the entire population (no randomness), then the site holds information.

The per-site entropy for site 18 is at neither extreme, but is entropically in the middle.

$$
\begin{aligned}
given: \quad & D = alphabet = \{A,T,C,G\} |D| = alphabet\ size = 4 \\
& p_A(18) = p_T(18) = 0 p_C(18) = p_G(18) = \frac{1}{2} \\
hence: H(18) &= -\sum_{d \in D}^{A,T,C,G} p_d(18) \log_{|D|} p_d(18) \\
&= -\left(0 \log_4 0 + 0 \log_4 0 + \frac{1}{2} \log_4 \frac{1}{2} + \frac{1}{2} \log_4 \frac{1}{2}\right) \\
&= -\left(0 + 0 + -\frac{1}{4} + -\frac{1}{4}\right) = \frac{1}{2}
\end{aligned}
$$

Intuitively, this states that if there is some entropy(randomness) in the samples of the site, then there is only partial information.

$$
\begin{aligned}
I(18) &= H_{\max}(18) - H(18) \\
&= 1 - \frac{1}{2} = \frac{1}{2}
\end{aligned}
$$

The complexity of a population(ensemble) S of sequences s, is defined from (14), as the maximal entropy of the population(ensemble) S minus the sum over the length ℓ of the per-site entropies.

$$
\begin{aligned}
&\textit{in (14) we found that} vC = H(S|\emptyset) - \sum_{i=1}^{|s|} H(i), \\
&\textit{in (10) we found that } H(S|\emptyset) = |s|\,, \\
&\textit{and in (15) we found that } |s| \equiv \ell
\end{aligned}
$$

$$
\begin{aligned}
\textit{hence} : C &= |s| - \sum_{i=1}^{|s|} H(i) \\
&= \ell - \sum_{i=1}^{\ell} H(i) \qquad (21)
\end{aligned}
$$

The equivalence of the maximum complexity to the length matches the intuitive understanding that, if a population of sequences of length ℓ has no redundancy, then their complexity is their length ℓ.

If G represents the set of all possible genotypes constructed from an alphabet D and of length ℓ, then the size(cardinality) of $|G|$ is equal to the size of the alphabet $|D|$ raised to the length ℓ.

$$|G| = |D|^{\ell} \qquad (22)$$

For the complexity measure to be accurate, a population sample size of $|D|^{\ell}$ is suggested to minimise the error[4, 8]. This quantity can be computationally unfeasible. For practical applications, Adami[5] chooses a population size of 3600 for an alphabet of size twenty eight, $|D| = 28$, and a length of one hundred, $\ell = 100$. This is a population size of about $1.29\,|D|\,\ell$. The reason it is greater than $|D|\,\ell$ is because the population size will fluctuate in the simulation, and it is necessary to maintain a minimum of $|D|\,\ell$ for statistical reliability of any trends present. So, for a population S, we choose with Adami a computationally feasible population size of $|D|\,\ell$, which is sufficient to show any trends present:

$$|S| \geq |D|\,\ell \qquad (23)$$

4.1.3 Epistasis in Physical Complexity

The complexity C in (21), based on summing the per-site entropies in (13), is only an estimate of the population complexity, as shown in (11), because the probability $p_d(i)$ in (16) of finding a character d at site i can be dependent on characters at other sites, not just the frequency of d at site i. Such correlations between sites are called epistatic. These epistatic effects can render the complexity significantly different from the sum of the per-site entropies[5].

The method proposed[5] for managing the epistatic effects involves calculating C_e for a population(ensemble) S in its environment e, which is the population complexity taking into account the epistatic interactions, rather than C which does not take into account epistatic effects. C_e is still the difference between the maximal entropy $H(S|\emptyset) = |s|$ and the actual entropy $H(S|e)$, as in (11), but $H(S|e)$ is not estimated from the per-site entropies as in (13). So the physical complexity of sequences s in the population S is $\langle K(s:e)\rangle_S$, which will be abbreviated to C_e, and is the maximal entropy $H(S|\emptyset)$ minus the actual entropy $H(S|e)$(specifically not estimated as the sum of the per-site entropies):

$$
\begin{aligned}
&in\ (11)\ we\ found\ that\ \langle K(s:e)\rangle_S = \sum_{s\in S} p(s)K(s:e) \approx H(S|\emptyset) - H(S|e),\\
&in\ (10)\ we\ found\ that\ H(S|\emptyset) = |s|\,,\\
&in\ (15)\ we\ found\ that\ |s| \equiv \ell\\
&and\ given\ \langle K(s:e)\rangle_S = C_e
\end{aligned}
$$

$$
\begin{aligned}
hence: \langle K(s:e)\rangle_S = C_e &= \sum_{s\in S} p(s)K(s:e) \approx H(S|\emptyset) - H(S|e)\\
C_e &= H(S|\emptyset) - H(S|e)\\
C_e &= |s| - H(S|e)\\
C_e &= \ell - H(S|e) \qquad (24)
\end{aligned}
$$

So the population complexity taking into account epistatic effects C_e, is the length ℓ of the sequences in the population minus the entropy in the population $H(S|e)$. $H(S|e)$ can be calculated, as found in (12), taking into account all epistatic effect, by the sum over the sequences s in the population(ensemble) S of the conditional probabilities $p(s|e)$, probability to find sequence s given the environment e, multiplied by the logarithm of the probability $p(s|e)$:

$$
in\ (12)\ we\ found\ H(S|e) = -\sum_{s\in S} p(s|e)\log p(s|e)
$$

In every finite population S, estimating $p(s|e)$ using the actual frequencies of s found in S (if those could be obtained) results in corrections to (12) larger than the quantity itself[8], rendering the estimate useless. Another avenue for estimating the entropy is to assume that the population S is of infinite size, at which point it would include all viable sequences, where viable means the sequence s is found in the environment e. Each viable sequence(genotype) would occur with equal probability, as it is an infinite population. The infinite population S can be reduced to the set of all viable sequences[5], denoted S_{viable}, a viable sequence denoted s_{viable}, and mutational clones can be created at several positions at the same time[5, 15, 26] to measure the epistatic effects. The mutational clones will be discussed later. As S_{viable} is a set of all viable sequences, each sequence s_{viable} occurs with equal probability within the set, as it did in the infinite population S. Therefore the probability of each sequence is one over the cardinality of S_{viable}.

$$
\begin{aligned}
given: \quad & (12),\ s_{viable} \in S_{viable},\ p(s_{viable}|e) = \frac{1}{|S_{viable}|} \\
hence: \ H(S|e) &= -\sum_{s_{viable} \in S_{viable}} p(s_{viable}|e) \log_{|D|} p(s_{viable}|e) \\
&= -\left(|S_{viable}| \left(\frac{1}{|S_{viable}|} \log_{|D|} \frac{1}{|S_{viable}|} \right) \right) \\
&= -\left(\log_{|D|} \frac{1}{|S_{viable}|} \right) \\
&= \log_{|D|} |S_{viable}| \qquad (25)
\end{aligned}
$$

To determine $|S_{viable}|$, a wild-type genotype(sequence) can be taken from the population sample. The wild-type is the genotype found to be in the majority, i.e. the highest frequency in the population. As this procedure involves an evaluation of fitness, it is easiest for organisms whose survival rate is closely related to their organic fitness: i.e., for organisms who are not epistatically linked to other organisms in the population[5]. The wild-type can be used to estimate the number of viable genotypes(sequences), $|S_{viable}|$ in the following manner[5]. The mutants of a wild-type that have a single mutation (change of agent in the agent-chain sequence) are said to be at mutational distance one. The number of mutations is the mutational distance n, so two mutations in the mutant would be at mutational distance 2 of the wild-type.

If we define $|S_{viable}(n)|$ as the number of viable(greater or equal fitness to the wild-type) genotypes(sequences) at mutational distance n, then we can define $w(n)$ as the proportion of viable genotypes at mutational distance n over the total number of mutants at the mutational distance n:

$$
\begin{aligned}
w(n) &= \frac{|S_{viable}(n)|}{|D|^n \ell!/((\ell-n)!n!)} \qquad (26) \\
where: \ & 1 \leq n \leq \ell \\
given: \ & \ell = length
\end{aligned}
$$

It is important to note that the set of mutants at mutational distance 2, $|S_{viable}(2)|$, includes all the mutants at mutational distance 1, $|S_{viable}(1)|$. Furthermore, in a population of sequences with the same length ℓ, the maximum mutational distance is the length ℓ of the sequences, after which further mutations would be repetitions of existing mutants. So in order to calculate $|S_{viable}|$ directly would require determining all the viable mutants at mutational distance ℓ, $|S_{viable}|$, by calculating $w(\ell)$, where ℓ is the length of the sequences in the population. This is equivalent to testing all $|D|^\ell$ sequences(genotypes), from (23), and is not computationally feasible. However, $w(\ell)$ can be estimated from $w(n)$ for small values of n, for $1 \leq n \leq 8$[5]. So (25) can be written in terms of $w(\ell)$ using (27) as follows:

$$
given: \qquad (24),\ (25),\ n = \ell,\ |G_{viable}| = |G_{viable}(\ell)|
$$

$$
\begin{aligned}
\textit{hence}: \qquad & w(\ell) = \frac{|S_{viable}(\ell)|}{|D|^{\ell}\, l!/((l-\ell)!\ell!)} = \frac{|S_{viable}(\ell)|}{|D|^{\ell}} \\
& |S_{viable}(\ell)| = w(\ell)\,|D|^{\ell} \\
\textit{and hence}: \; H(S|e) &= \log_{|D|}\left|S_{viable(\ell)}\right| \\
&= \log_{|D|}\left(w(\ell)\,|D|^{\ell}\right) \qquad (27)
\end{aligned}
$$

Even for small values of n, the number of mutants that need to be created to calculate $w(n)$ can be very large, but as it is a proportional value, random sampling can be performed to calculate $w(n)$ effectively, at least for small values of n. This has been proven experimentally by Adami[5], and we will check this further with our simulation. The results of the $w(n)$ population sampling can be used to estimate $w(\ell)$, by fitting a curve to the results. The results are fitted well by the following two parameter equation[5, 26]:

$$
\begin{aligned}
w(n) &= |D|^{-\alpha n^{\beta}} \qquad (28) \\
where: \; & 0 \leq \alpha \leq 1
\end{aligned}
$$

In biological terms, $(1-\alpha)$ measures the degree of neutrality in the code and β reflects the degree and form of epistasis. If $\beta > 1$, each successive mutation tends to reduce fitness more than previous ones (synergistic epistasis), and if $\beta < 1$, each additional mutation is progressively less damaging on average (antagonistic epistasis).

By sampling mutants at computationally feasible mutational distances, and thereby calculating the $w(n)$ values, it is proposed to calculate α from $w(1)$ directly, for which the sample space can be fully searched. Then the higher $w(n)$ values can be used to calculate β[5, 26]. However, better values for α and β can be calculated in general by minimising a least squares fit:

$$
\sum_{1}^{n}\left(|D|^{-\alpha n^{\beta}} - w(n)\right)^2 \qquad (29)
$$

So $w(\ell)$ in (27) can be expressed in terms of α and β using (28), and therefore the population complexity C_e in (24) can also be expressed in terms of α and β.

$$
\begin{aligned}
& \textit{in (24) we found that } C_e = \ell - H(S|e), \\
& \textit{in (27) we found that } H(S|e) = \log_{|D|}\left(w(\ell)\,|D|^{\ell}\right), \\
& \textit{in (28) we found that } w(n) = |D|^{-\alpha n^{\beta}}, \\
& \textit{and given } n = \ell
\end{aligned}
$$

$$
\textit{hence}: \; w(\ell) = |D|^{-\alpha \ell^{\beta}}
$$

$$
\begin{aligned}
\textit{and hence}: \; C_e &= \ell - H(S|e) \\
C_e &= \ell - \log_{|D|}\left(w(\ell)\,|D|^{\ell}\right) \\
C_e &= \ell - \log_{|D|}\left(|D|^{-\alpha\ell^{\beta}}\,|D|^{\ell}\right) \\
C_e &= \ell - \left(\log_{|D|}|D|^{-\alpha\ell^{\beta}} + \log_{|D|}|D|^{\ell}\right) \\
C_e &= \ell - \left(-\alpha\ell^{\beta} + \ell\right) \\
C_e &= \alpha\ell^{\beta} \qquad (30)
\end{aligned}
$$

This complexity C_e takes into account all epistatic interactions, but it has strict requirements for its application. Most importantly, the wild-type sequence used for creating the mutational clones must not be epistatically linked to other sequences in the population.

4.2 Physical Complexity Extended

Reformulating the complexity measure for a population of agent-chains requires consideration of the following issues; mapping of the sequence sites to the agent-chains, managing a population of variable length sequences, the creation of a performance measure, and clustering within populations

4.2.1 Mapping Sequence Sites

The first concern is about mapping the agent-chains to sites. The Physical complexity measure has been applied to DNA sequences and populations of self-replicating programs in the artificial life simulator Avida[2]. For the DNA, the sites were the nucleotides from which it is constructed, and for the artificial life simulator the sites were the program instructions which made up the self-replicating programs. The intuitive answer is that as the agents are the functional unit for processing, the sites map to them. However, there are alternatives. All the possible mappings are listed below:

(i) the program instructions of the agents' executable components
(ii) the descriptions of the agents (BML)
(iii) the agents themselves

The program instructions of (i) could be easily mapped to the sites of the sequence, with there being similar work of Physical complexity applied to populations of self-replicating programs in the artificial life simulator, Avida. However, the Avida platform consisted of far more simplistic sequences(programs) than those proposed in the MAS model. These individuals were all composed from the same set of atomic instructions. The MAS model has no such constraint that the executable components of the different agents be in the same programming language, only that they have compatible interfacing(SDL). It would be unfortunate to have to create such a constraint.

The descriptions (BML) (ii) are comparable to one another, so it does not suffer from the same problem as (i). The descriptions describe one or more processes, and the attributes

associated with each. If the mapping is at the level of the process and each agent performs only a single process, the mapping would be equivalent to mapping to the level of the agent(iii). If the process is parameterised based on its attributes, then the parameters will map to the sites. The parameters exist as an unordered set, and to apply the Physical complexity measure would require converting them into an ordered list. This would be difficult or impossible, and therefore (ii) not preferred.

Mapping intuitively to the level of the agent (iii) avoids the problems of both (i) and (ii), as the agent is the base unit for evolution in the population of the MAS model. Unfortunately, it will create the constraint that the agents are atomic, such that each agent performs only a single process. The atomicity requires that no single process agent can functionally replace any sequence of single process agents of length two or more. However, this constraint does not prevent an agent from representing multiple agents in a chain, provided that it is done openly, defined in the agents description. Mapping the site for entropy calculation to the level of the agent is the most preferable choice, as it creates the least restrictive constraints. The problem of the atomicity will be considered further in Section 4.2.5. For the subsections in between it will be assumed that agents are atomic.

4.2.2 Physical Complexity for Variable Length Populations(VLPs)

Managing populations of variable length agent-chains is the most significant part of the reformulation, because unlike the other issues which are extensions of the measure, this requires changing and re-justifying the fundamental assumptions.

It is necessary to understand the measures conditions and limits, in terms suitable for extending the measure to variable length populations. The Physical Complexity C is defined for a population in which the individuals all have length ℓ:

$$in\ (21)\ we\ found\ that\ C = \ell - \sum_{i=1}^{\ell} H(i)$$

Quite simply what does the length ℓ equal if the population is of variable length? The problem is what ℓ represents, which is the **complexity potential** C_p, the maximum complexity possible for the population. The maximum complexity occurs when the per-site entropies sum to zero, as there is no randomness in the sites (all contain information). So:

$$if \sum_{i=1}^{\ell} H(i) \to 0\ in\ (21)\ then\ C \to \ell \tag{31}$$

The complexity potential equals the length, $C_P = \ell$, provided the population S is of sufficient size $|S|$ for accurate calculations, i.e. $|S|$ is equal or greater than $|D|\,\ell$, as found in (23), justified before at the end of section 4.1.2.

$$in\ (23)\ we\ found\ that\ \ |S| \geq |D|\,\ell$$

For a variable length population(VLP) S, the complexity potential C_{V_P} cannot be equivalent to the length ℓ, because it does not exist. The C_{V_P} is zero, when the population size is less than the alphabet size, $|S| \leq |D|$. If $|S| \geq |D|$, then there are at least $|D|$ individual samples of length one or more, so the C_{V_P} will be greater than zero and will be equivalent to the length of a VLP ℓ_V, which will be defined shortly.

$$\begin{aligned} C_{V_P} &= \begin{cases} 0 & if\ \ |S| < |D| \\ \ell_V & if\ \ |S| \geq |D| \end{cases} \\ where &: \quad \ell_V \geq 1,\ |D| > 0 \\ given &: \quad S = population,\ D = alphabet \end{aligned} \tag{32}$$

If ℓ_V where to be equal to the length of the longest individual ℓ_{max} in the VLP S, then the operational problem is that for some of the later sites i in the range of one to ℓ_{max}, the number of samples per-site will be less than the population size. So the complexity potential $C_{V_P} = \ell_V = \ell_{max}$ would be incorrect, as there are insufficient samples at the later sites. Consider the samples of DNA sequences presented earlier, but with different samples for three and four.

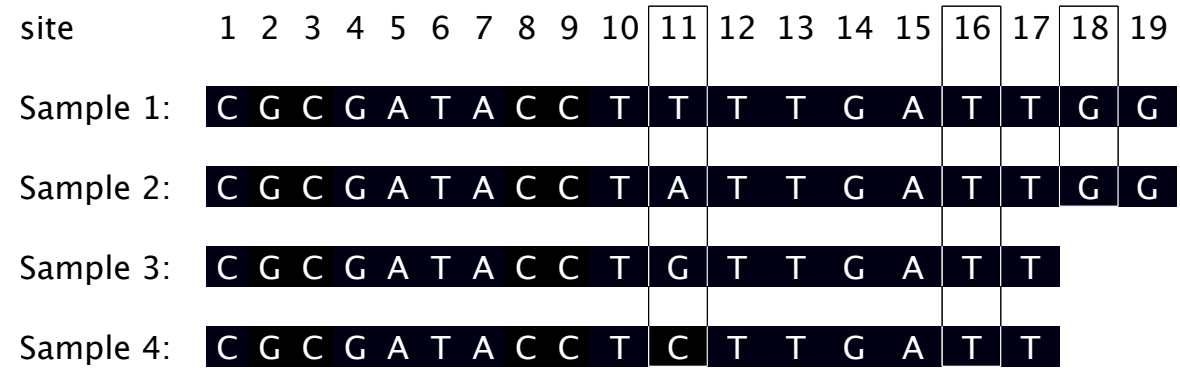


Figure 5: Genome Samples

If the entropy and information are calculated for site 18, $I(18) = 1$, it is significantly different to the values calculated earlier, $I(18) = \frac{1}{2}$. So, There must be a sufficient sample size for the per-site entropy calculations, so as to be able to accurately estimate the $p_d(i)$ probabilities. Therefore the length ℓ_V for a VLP is defined as the highest value within the range between one and the maximum length $1 \leq \ell_V \leq \ell_{\max}$, for which there are sufficient samples to calculate the complexity. To specify the value of ℓ_V precisely, the following function $sampleSize(i : site)$ is required, which provides the number of samples at a given site:

$$\begin{gathered} sampleSize(i : site) output : int \\ where :\ 1 \leq output \leq |S|\,,\ |S| = population\ size \end{gathered}$$

Therefore the length ℓ_V for a VLP is defined as the highest value within the range between one and the maximum length $1 \leq \ell_V \leq \ell_{\max}$, for which the number of samples at a site specified by ℓ_V is greater than or equal to the alphabet size $|D|$ multiplied by length ℓ_V:

$$sampleSize(\ell_V + x) \leq sampleSize(\ell_V) \geq |D|\,\ell_V \tag{33}$$

$$\begin{aligned} where: \quad & 1 \leq \ell_V \leq \ell_{max} 0 < x < \ell_{max} - \ell_V \, |S| \geq |D| > 0 \\ given: \quad & D = alphabet,\ \ell_V = length\ for\ a\ VLP, \\ & S = population,\ \ell_{max} = maximum\ length\ in\ a\ VLP \end{aligned}$$

This definition intrinsically includes the minimum population size for VLPs, $|D|\,\ell_V$. This replaces the minimum population size for same length populations as specified in (23).

C_{V_P} in (32) will always equal ℓ_V provided the condition $|S| \geq |D|$ is true. If it is false, the C_{V_P} will be zero and therefore the complexity of the VLP S will be zero, so no calculation involving ℓ_V will be performed.

The length ℓ in (15) no longer exists, so the per-site entropy calculation (16) must be updated, so the per-site entropy calculation for VLPs will be denoted by $H_V(i)$. It remains algebraically almost identical to (16), but the conditions an constraints of its use will change, specifically ℓ will be replaced by ℓ_V:

$$H_V(i) = -\sum_{d \in D} p_d(i) \log_{|D|} p_d(i) \tag{34}$$

$$\begin{aligned} where: \quad & 1 \leq i \leq \ell_V,\ 0 \leq p_d(i) \leq 1,\ \sum_{d \in D} p_d(i) = 1 \\ given: \quad & i = site,\ \ell_V = length\ for\ a\ VLP, \\ & D = alphabet,\ |D| = alphabet\ size \end{aligned}$$

$H_V(i))$ in (34) ranges between 0 and 1, as $H(i))$ in (16) did. The condition on the site i changes from ℓ at the upper limit to ℓ_V.

$$\begin{gathered} 0 \leq H_V(i) \leq 1 \\ where:\ 1 \leq i \leq \ell_V \end{gathered} \tag{35}$$

As before in (17), if a site i is identical across the population, it will have no entropy.

$$\begin{gathered} H_{V_{min}}(i) = 0 \\ where:\ 1 \leq i \leq \ell_V \end{gathered} \tag{36}$$

Similarly as in (20), a site i has maximum entropy if the $p_d(i)$ probabilities in (34) are equal. In effect the content of the site is uniformly random. This is true for all i:

$$H_{V_{max}}(i) = 1 \qquad (37)$$
$$given: \ p_d(i) = \tfrac{1}{|D|} \qquad where: \ 1 \leq i \leq \ell_V$$

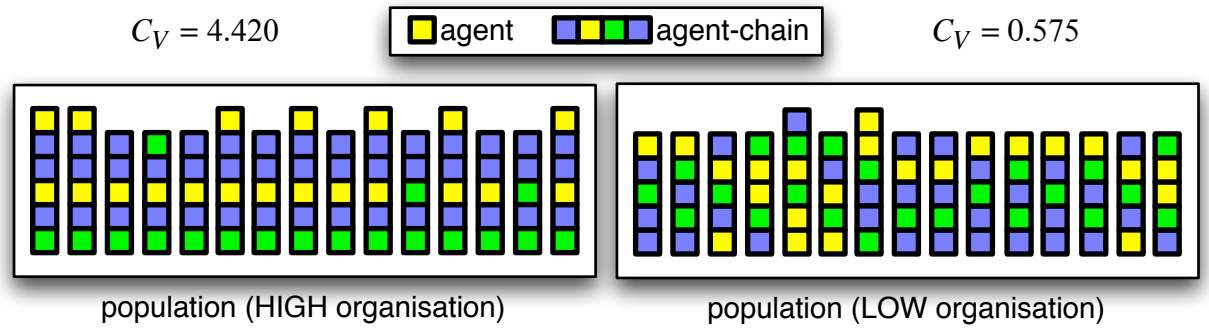
Similarly as in (20), when the entropy is minimum, zero, then the site i holds information, as every sample shows the same character of the alphabet. When the entropy is maximum, the character found in the site i is uniformly random and therefore holds no information. Therefore the amount of information is the maximal entropy of the site (38), minus the actual per-site entropy (34):

$$I_V(i) = H_{V_{\max}}(i) - H_V(i) = 1 - H_V(i) \qquad (38)$$
$$where: \ 1 \leq i \leq \ell_V$$

The complexity C_V of a variable length population(VLP) is the complexity potential of the VLP, C_{V_P}, minus the sum over the length of the VLP ℓ of the per-site entropies:

$$C_V = \begin{cases} 0 & if \ C_{V_P} = 0 \\ \ell_V - \sum_{i=1}^{\ell_V} H_V(i) & if \ C_{V_P} > 0 \end{cases} \qquad (39)$$
$$where: \ \ell_V = length \ for \ VLP, \ i = site, \ H_V(i) = VLP \ entropy \ for \ a \ site \ i$$

Now that the Physical complexity can be applied to VLPs, and the site mapping has been resolved, it can used to measure the organisational complexity of agent-chains populations shown in Figure 6, where l_V is calculated from (33):



$$D = alphabet = \{\blacksquare, \blacksquare, \blacksquare\}$$
$$|D| = alphabet \ size = 3 \quad \ell_{max} = maximum \ length = 6$$
$$\ell_V = length \ for \ a \ VLP = 5 \quad complexity \ potential \ C_{V_P} = \ell_V = 5$$

Figure 6: Organisational Complexity in Populations of Agent-chains

If we recall the definition of **organisational complexity**(organisation), which measures the complexity of the coherent patterns and structures, formed by the clustering of the agents within the population of agent-chains. The Physical complexity values match the intuitive understanding, that one gets visually, about the organisational complexity in the populations.

4.2.3 Performance Measure

Based on the Physical Complexity measure of organisation, a **performance measure** can be constructed which shows the use of the information space, i.e. the organisational complexity relative to the maximum possible organisational complexity. The efficiency E for a VLP can be stated as the actual complexity C_V over the complexity potential C_{V_P}:

$$E = \frac{C_V}{C_{V_P}} \tag{40}$$

The efficiency E ranges between zero and one:

$$0 \leq E \leq 1 \tag{41}$$

The $\%E$ percentage efficiency can be taken to make the results more readable, ranging between zero and a hundred:

$$0 \leq E \leq 100 \tag{42}$$

The maximum efficiency E_{max} occurs when the actual complexity C_V equals the complexity potential C_{V_P}, when there is no randomness in the population.

$$E_{max} = 1 \tag{43}$$

The agent-chain populations considered earlier, for which the complexity measure C_V was applied are shown below with their respective efficiencies:

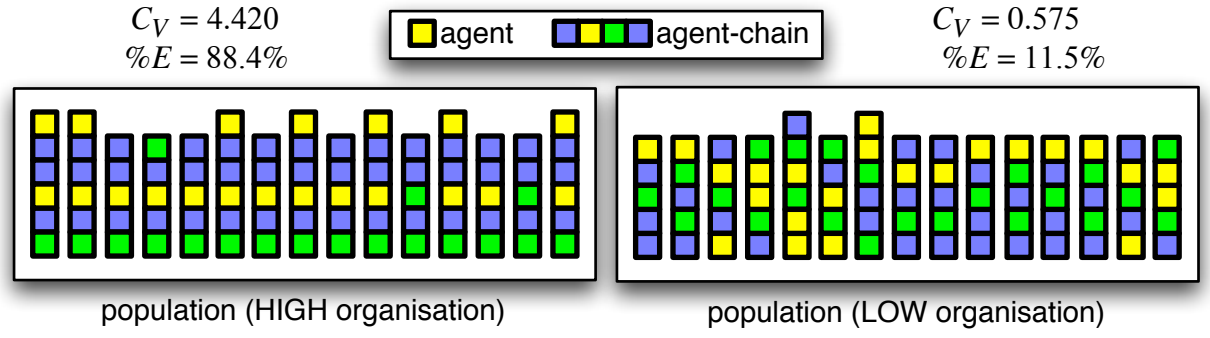


Figure 7: Efficiency in Populations of Agent-chains

The efficiency of the populations is as expected. It shows the population with high organisational complexity is efficient, but that there is still some randomness in the population. For the population with low organisational complexity, it shows that the population is almost entirely random.

The performance measure is concise, succinct, and matches ones intuition.

4.2.4 Clustering

Clustering in a population of evolving agent-chains is the grouping of same or similar sequences around an optimum genome on the fitness landscape. If the evolutionary process does not become trapped at local optima, then there is only global optima. It is important to realise that the fitness landscape is the combination space of the agents weighted with their fitness values. As such it is dependent on the set of agents(alphabet) from which solutions are constructed as much as it is dependent on the selection pressure (fitness function).

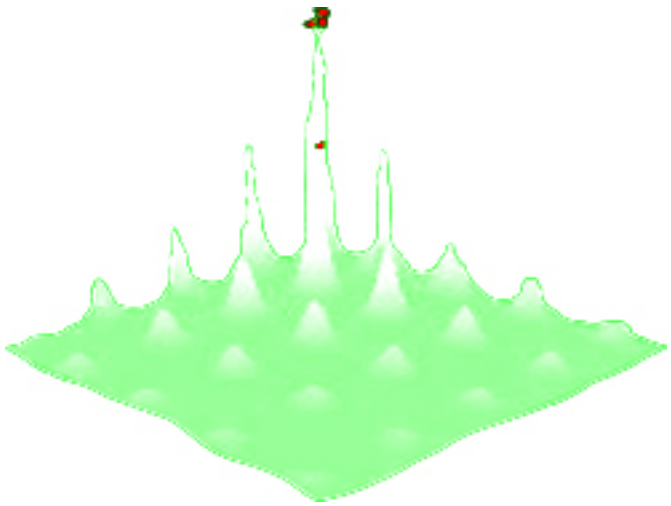


Figure 8: Fitness Landscape - Single Global Optimum

The population will move across the fitness landscape, temporarily clustering over local optima, on the way to clustering around the optimal genome at the peak of the global optimum, assuming that the evolutionary process does not become trapped at local optima. The population will eventually cluster over all global optima. In Figure 8 the population will cluster into a single cluster at the peak of the global optimum. This process of clustering at the peak of a global optimum is indicated by the the average population fitness F_{avg} tending towards the maximum fitness F_{max}, the **clustering indicator**. The optimal solution(sequence) which has maximum fitness F_{max} is increasingly becoming the dominant agent-chain(sequence) in the population.

$$F_{avg} \rightarrow F_{\max} \tag{44}$$

Later it will be shown that the clustering indicator is especially useful as it indicates the occurrence of clustering independent of the number clusters in the population. Also its

output is much more reliable to measure and programme than the clustering coefficient which is about to be introduced. This will be discussed further in section (7.3.3), Conclusions.

At the same time, the population complexity C_V (39) tends towards the complexity potential C_{V_P} (32), because the uniformity of sites across the population is increasing as the optimal solution(sequence) is increasingly becoming the dominant sequence in the population:

$$\begin{aligned} given: &\quad (40),\ C_V \to C_{V_P} \\ hence: &\quad E = \frac{C_V}{C_{V_P}} \to 1 \\ so: &\quad E \to 1 \wedge |T| \to 1 \\ where: &\quad |T| = number\ of\ clusters \end{aligned} \tag{45}$$

The efficiency E will not quite reach one due to mutations. The efficiency E acts as **clustering coefficient**, tending towards its maximum when the population consists of only one cluster, $|T| = 1$. The other extreme is when the number of clusters equals the size of the population, $|T| = |S|$. This would only occur with a non-discriminating selection pressure, in which the fitness landscape would be perfectly flat, as shown in Figure 9 below:

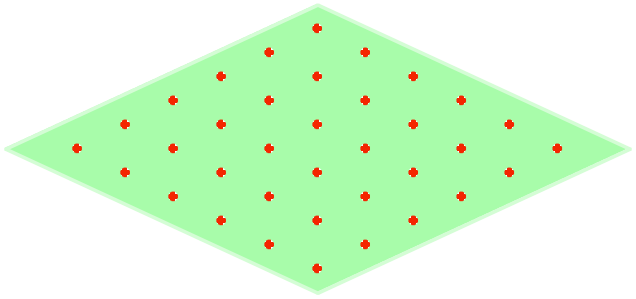


Figure 9: 3D Fitness Landscape - Perfectly Flat

The population occupancy of the fitness landscape would be uniformly random, as any position(agent-chain combination) has the same fitness for any agent-chain in the population. So the entropy(randomness) would be tending towards maximum, resulting in the complexity C_V (39) tending towards zero. It would never quite reach zero, due to mutation. The clustering indicator (44) would instantaneously be showing clustering, as the average fitness is always the maximum fitness, $F_{avg} = F_{max}$. As $C_V \to 0$, the clustering coefficient, efficiency E, would tend to zero.

$$\begin{aligned} given: &\quad C_V \to 0(40) \\ hence: &\quad E = \frac{C_V}{C_{V_P}} \to 0 \\ so: &\quad E \to 0 \wedge |T| \to |S| \end{aligned} \tag{46}$$

So the number of clusters $|T|$ would tend to the population size $|S|$. Each cluster would in fact consist of only one agent-chain.

If there are multiple global optima, the clustering indicator $F_{avg} = F_{max}$ behaves as before, because it is independent of the number of clusters.

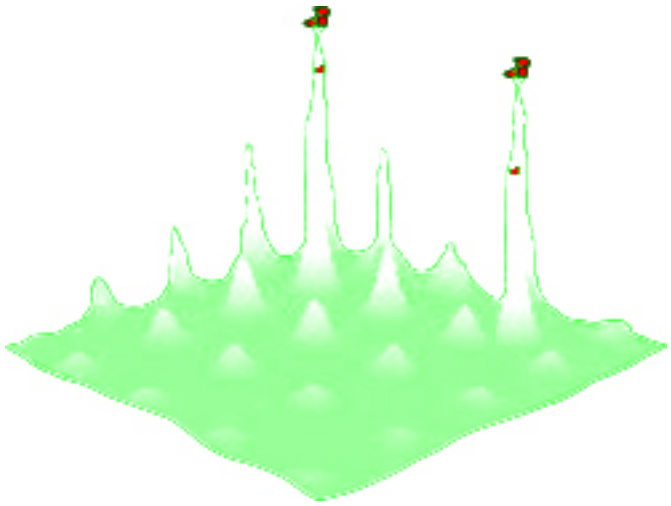


Figure 10: 3D Fitness Landscape - Multiple Global Optima

However, the efficiency E will no longer tend towards its maximum (45), precisely for the reason that the population consists of more than one cluster. For a population S, a cluster is a sub-population with an efficiency that tends towards the maximum (45). If T is the set of clusters, then its size $|T|$ is the number of clusters in the population.

The simplest scenario of multiple clusters, is when there are pure clusters. Pure meaning that there are no agents shared between the clusters. So, there are as many totally distinct optimal solutions with maximum fitness F_{max}, as there are clusters. In this scenario, the value the efficiency E no longer tends towards one, but a value based on the number of clusters $|T|$, because a *number* of the probabilities $p_d(i)$ in (34) at each site become one over the number of clusters $|T|$. The *number* of the probabilities is equal to the number of clusters. So:

$$
\begin{aligned}
\textit{given}: \quad & p_d(i) = \frac{1}{|T|},\ (32),\ (34),\ (40) \\
\textit{hence}: \quad & H_V(i) = -\sum_{d \in D} p_d(i) \log_{|D|} p_d(i) = -\sum_{d \in D} \frac{1}{|T|} \log_{|D|} \frac{1}{|T|} \\
& H_V(i) = -\left(|T| \left(\frac{1}{|T|} \log_{|D|} \frac{1}{|T|} \right) \right) = -\log_{|D|} \frac{1}{|T|} \\
& H_V(i) = \log_{|D|} |T| \\
\textit{and hence}: \quad & C_V \to \left(\ell_V - \sum_{i=1}^{\ell_V} \log_{|D|} |T| \right) \\
& E = \frac{C_V}{C_{V_P}} \to \frac{\ell_V - \ell_V(\log_{|D|} |T|)}{\ell_V} \\
& E \to 1 - (\log_{|D|} |T|) \qquad (47)
\end{aligned}
$$

With pure clusters, the value to which the clustering coefficient E is tends towards, can be used to determine the number of clusters $|T|$.

For clusters that are not pure, the relationship cannot be specified so succinctly. Environments with multiple optima, in the fitness landscape, will potentially lead to the population clustering around each global optimum, in which case the the efficiency E will no longer tend towards the maximum. For a population S with clusters, each cluster is a sub-population with an efficiency that tends towards the maximum. To specify this more accurately, we require the following function, $efficiency(input : population)$, which provides the efficiency E, as defined in (40), of the input population:

$$efficiency(input : population) output : int$$
$$where : \ 0 \leq output \leq 1$$

Assuming that the clustering indicator is actively indicating clustering, the clusters T of the population S can be defined as sub-populations of S, where all of them have efficiency E tending towards the maximum.

$$hence : \ t \in T \rightarrow \left(t \subseteq S \wedge efficiency(t) \rightarrow 1 \wedge |t| \approx \frac{|S|}{|T|} \wedge \sum_{t \in T} |t| = |S| \right) \tag{48}$$
$$given : \ S = population \wedge F_{avg} \rightarrow F_{\max}$$

As T is the set of clusters in the population S, then a cluster t in T, is a subset (sub-population) of the population S, and t has an efficiency E tending towards the maximum, one (43). Furthermore, the cluster size $|t|$ is roughly equal to the population size $|S|$ divided by the number of clusters $|T|$. It is only roughly equal, as the division may not result in a whole number. These conditions are true for all members of the set of clusters T, and the summation of the cluster sizes in T equals the size of the population $|S|$. This last condition ensures that clusters are non-overlapping, i.e. do not share agent-chains(members).

If we visualise the population on the 3D fitness landscape of Figure 10, in Figure 11, even though the clustering indicator is active, the clustering cannot be seen in the visualisation of the population.

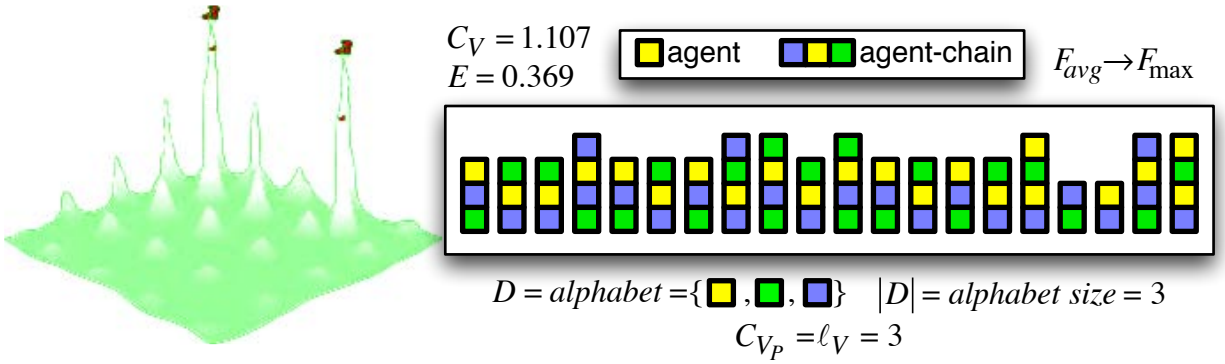


Figure 11: Population with Hidden Clusters

If we arrange the population to show the clustering, then we can clearly see the two clusters present in the population, in Figure 12.

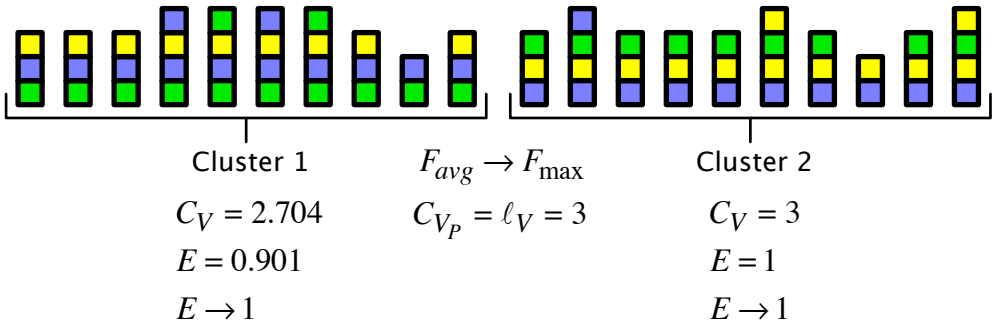


Figure 12: Population with Clusters Showing

Figure 12 clearly shows that the clusters in the population both have efficiencies E tending towards their maximum, compared to the efficiency E of the population as a whole which is tending towards a value significantly below the maximum. This is the behaviour of clusters as specified in (48).

The population size $|S|$ in Figures 11 and 12 is double the minimum requirement, specified in (33), so that the complexity C_V in (39) and efficiency E in (40) could be applied to the clusters without introducing new definitions, while simultaneously explaining the basic principles of clustering. However, when determining the variable length ℓ_V of a cluster t the sample size requirements are different, specifically a cluster t is a sub-population of S, and therefore by definition cannot have a population size equivalent to S (except when the population consists of only one cluster). Therefore (33) must be updated to manage clusters:

$$\ell_V = \begin{cases} sampleSize(\ell_V + x) < sampleSize(\ell_V) \geq |D|^{\ell_V} & if\ for\ population\ S \\ sampleSize(\ell_V + x) < sampleSize(\ell_V) \approx \frac{|D|\ell_V}{|T|} & if\ for\ cluster\ t \end{cases} \tag{49}$$

$$\begin{aligned} where: \quad & 1 \leq \ell_V \leq \ell_{max},\ 0 < x < \ell_{max} - \ell_V \\ & |S| \geq |D| > 0,\ t \in T,\ |T| = number\ of\ clusters \end{aligned}$$

$$\begin{aligned} given: \quad & D = alphabet,\ \ell_V = length\ for\ a\ VLP \\ & S = population,\ \ell_{\max} = maximum\ length\ in\ a\ VLP \\ & T = set\ of\ clusters\ in\ S \end{aligned}$$

A population with multiple clusters will always have an efficiency E that never tends towards the maximum. A reformulation is required of the efficiency measure E, to an efficiency measure capable of managing populations with multiple clusters E_m. It is equivalent to E if the population consists of only one cluster, and if there are multiple clusters E_m is the average of the efficiencies of the clusters:

$$E_m(S) = \begin{cases} \frac{C_V}{C_{V_P}} & if\ |T| = 1 \\ \frac{\sum_{t \in T} E_m(t)}{|T|} & if\ |T| > 1 \end{cases} \tag{50}$$

$$\begin{aligned} where: \quad & S = population \\ & T = set\ of\ clusters\ in\ population\ S\ as\ defined\ in\ (47) \\ & |T| = number\ of\ clusters\ in\ population\ S \end{aligned}$$

Different clusters in a population are not necessarily different species. This depends on the definition of species within the DBE and hence the MAS model. In biology, different species are often defined by the lack of ability to interbreed. This is represented in evolutionary computing by crossover not being able to be performed.

4.2.5 Atomicity

The property of atomicity in the agent pool, which is the set of available agents at the agent station, requires no agent to be able to functionally replace an agent-chain with a length of two or more.

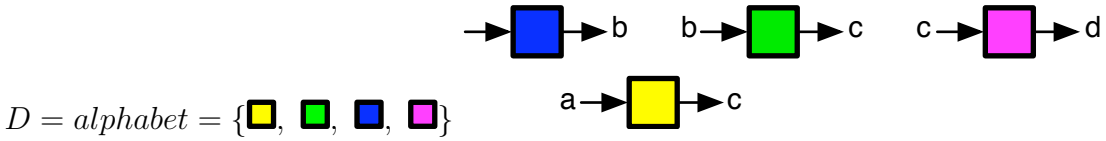


Figure 13: Non-atomic Agents

This requirement was necessary, because non-atomic agents adversely affect the uniformity calculated in the per-site entropies, which ultimately indicates information. So, information is being lost. Consider the following simple example:

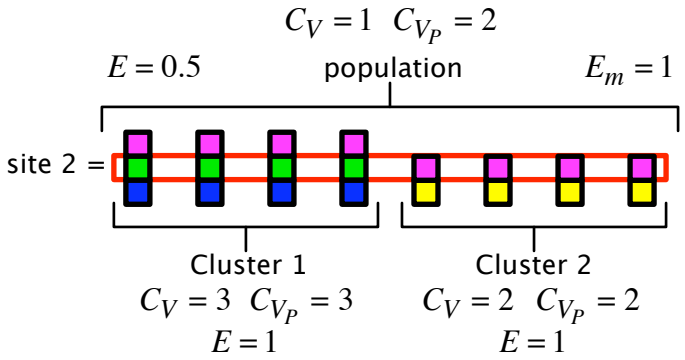


Figure 14: Population including Non-atomic Agents

As functionally replaces , the uniformity across site two is lost, shown in Figure 16. The efficiency E of the population is 0.5 rather than 1, because of the non-atomicity. The clustering defined in the previous subsection is vital in managing the non-atomicity, because the non-atomicity leads to the formation of clusters. These clusters can be found, indicated by the clustering indicator (44) and clustering coefficient (45), and the

efficiency for populations with multiple clusters E_m (50) can be used to calculate the actual efficiency, which in this case is one, as shown in Figure 16.

It should be noted that clusters formed by the presence of non-atomic services cannot be compared by their complexity values, C_V (39), even though they are functionally identical, i.e. perform the same task. Due to the non-atomicity they have different complexity values C_V, as shown in Figure 16. The complexity C_V is an absolute measure, whereas the efficiency E (40) is a relative measure. So, the clusters can be compared by their efficiencies, as shown in Figure 16.

With the efficiency measure E_m (50) for populations with multiple clusters, and with non-atomicity fitting the clustering model, atomicity is no longer required in the agent pool for the application of the organisational complexity measure, Physical complexity.

4.3 Relation to Fitness

Now that organisation, organisational complexity (39), has been defined, its relation to fitness can be considered. Fitness is a property of individuals, not of the population. The average fitness, although a population measure, does not measure the organisation(clustering).

4.3.1 Original Physical Complexity

The maximum fitness is part of the evolutionary process and increases over the generations. The maximum fitness never falls as the selection pressure is static and the mutation rate is not high enough to cause the loss of all the maximum fitness solutions from the population[5].

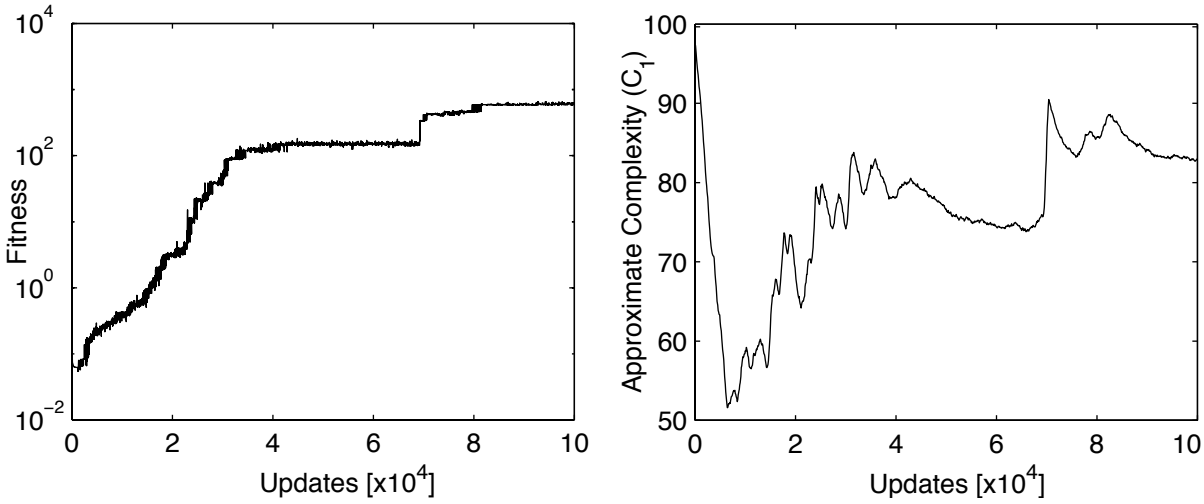


Figure 15: Physical Complexity & Fitness Graphs[3]

The complexity increases over the generations, but can suffer short-term falls due to the arrival of fitter mutants. As new fitter mutants arrive and spread throughout the population over several generations, the uniformity of the sites will fall, increasing the randomness, which is measured in the per-site entropies. Therefore, the complexity,

which is the complexity potential minus the sum of the per-site entropies, decreases temporarily[5].

The complexity initially starts high, which is due to the population being seeded with a single sequence (individual), that temporarily takes over the population[3].

4.3.2 Physical Complexity for VLPs

The Physical Complexity for VLPs in (39) has the same structure and properties as the original Physical complexity in (21), only the input has changed, so the relation between fitness and complexity is the same.

The further work using fitness as a clustering indicator in (44), can be summarised as follows. The convergence of the average fitness to the maximum fitness indicates clustering within the population (44), and the efficiency (40) acts as clustering coefficient, showing if there are one or more clusters. In the case of pure clusters, the clustering coefficient can show the exact the number of clusters (47).

5 Methods

The relevant assumptions and requirements for the simulation to effectively answer the proposed hypotheses will be presented in this section.

5.1 Simulation

The core assumptions and the abstract model from Section 2 are the primary requirements for the simulation. Additional detailed assumptions are specified in the following subsections.

Concerns that the simulation does not accurately or appropriately represent the Evolutionary Environment(EvE), proposed to be part of the DBE core architecture[17], would be unfounded. This is because the initial prototypes of this simulation were used to create the EvE proposed in Evolutionary Environment Discussion Paper[10]. Furthermore, the core structure of the simulation has not been changed since it was included in the discussion paper, except for allowing the running of a single population rather than an entire ecosystem, and the addition of methods for calculating the organisational complexity.

5.1.1 Requirements

The evolutionary computing process at the simulated agent stations must work as expected, any failures will be removed during testing. The following requirements must be met:

- reaches global optimum/optima
  (i.e. does not get trapped at local optima - non-trapping fitness function)
- maximum fitness never decreases
  (given a normal/low mutation rate and a fixed selection pressure)
- average fitness tends towards maximum fitness during clustering
  (as this is the clustering indicator)

If the evolutionary computing process is encoded faithfully into the simulation, with a normal/low mutation rate, a fixed selection pressure and a non-trapping fitness function, then these requirements will be intrinsically present. They are not additional tasks to be done, which artificially weight the simulation, biasing it towards the desired or expected results. If during the testing these properties are not present, there must be error in the simulation, which will have to be debugged.

5.1.2 Class Diagram

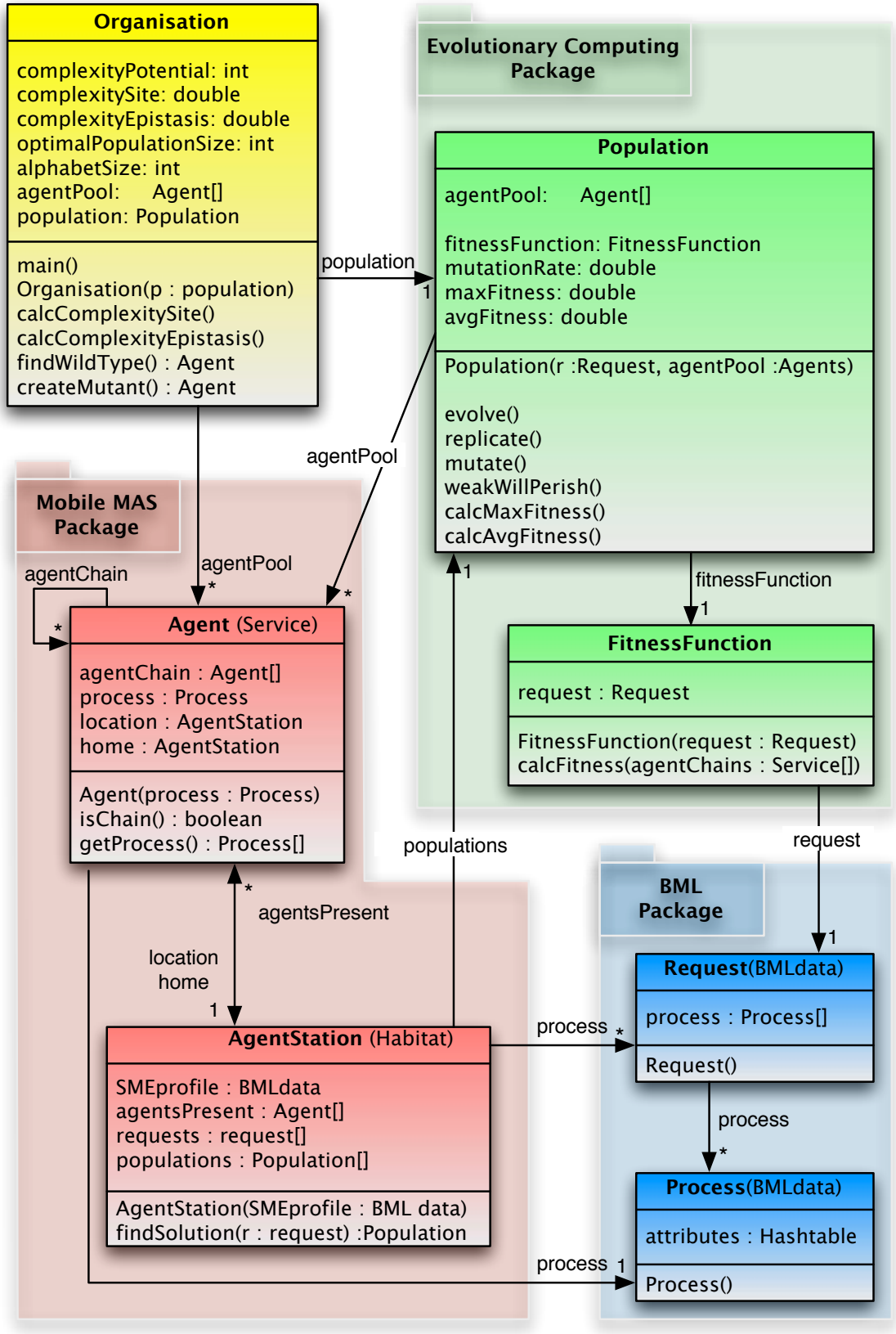


Figure 16: Simulation Class Diagram

5.1.3 Simulation Setup

BML Package
This package consists of an abstraction of BML[12] for matching agents to user requests[10]. The Process class represents a business process, and contains a list of integer tuples in the attributes object. Each tuple represents an attribute of the business process, one integer for the identifier, and one for the value of the attribute. The attribute identifiers, range in value between one and ten, and in quantity between three and five. The values of the attributes range between one and a hundred. A sample is shown below:

$$Process = BML\ Business\ Process = [(1, 40), (5, 67), (3, 88)]$$

Figure 17: BML Business Process

A user request is represented by the Request class, which stores a business process request. The request consists of a list of BML business processes, stored as an array of Process objects. A sample is shown below:

$$Request = UserRequest = [[(3, 91), (4, 15)], [(8, 57), (6, 45), (3, 88)], [(1, 40), (5, 67)]]$$

Figure 18: User Request

A Request object consisting of 20 Process objects is instantiated at runtime to represent a complex user request. This request remains unchanged throughout the simulation.

Mobile MAS Package
This package represents a Multi-Agent System with mobile agents that can move between the agent stations, similarly to organisms that migrate between habitats in an ecosystem (Appendix A), the same as how DBE SM migrate between Habitats in the Evolutionary Environment[10]. The Agent class represents an agent(DBE service), which consists of much information, most importantly it contains a process object to represent its BML description. So, both the user request and agents have BML components which can be compared in the evolutionary process by the fitness function.

The AgentStation class represents a location at which an agent can move to, and has been simplified for the simulation to provide a single population. Specifically, it provides an alphabet(agent pool) for the evolutionary process, and a single population to solve the user request. It is instantiated with a set of 15 randomly generated agents.

Evolutionary Computing Package
This package implements a simple evolutionary process. The FitnessFunction class is instantiated with a Request object (user request) to determine the fitness of the solutions in the population. This is done by comparing the Process objects of the agent-chains with the list of Process objects in the Request object (user request). The comparison calculates the distance as a percentage value (real number), between the agent-chains and

the user request with respect to their Process(BML) objects. So, the FitnessFunction assigns fitness values between 0.0 and 100.0 to each agent-chain in the population.

The Population class represents the Population Objects in the Evolutionary Environment[10]. The Population class is instantiated to find a solution to the instantiated Request class (user request), from the set of agents instantiated in the AgentStation object. The solution is found by evolving the population of solutions, which are initially created by copying the set of agents from the AgentStation object to the Population object.

The evolutionary process works by assigning fitness values to the current population using the FitnessFunction object. Then the bottom 10% of the population is deleted (death rate), and then the top 10% of the population is allowed to replicate (replication rate). Subsequently, 10% of the population is mutated randomly using point mutations (mutation rate), with agents drawn from the set of agents in the AgentStation object. The point mutations consists of insertions (an agent is inserted into an agent-chain), replacements (an agent is replaced in an agent-chain), and deletions (an agent is deleted from the agent-chain).

The population size $|S|$ is maintained at $1.2\,|D|\,\ell_V$, to ensure (33) remains true, so that the organisational complexity measure can be calculated, Physical complexity for VLPs C_V in (39). This is done by temporarily increasing the replication rate or death rate. If the population size falls below $1.2\,|D|\,\ell_V$, then the replication rate is increased to 11% until the population size returns to $1.2\,|D|\,\ell_V$, at which point it is dropped back to 10%. If the population size increases above $1.2\,|D|\,\ell_V$, then the death rate is increased to 11% until the population size decreases to $1.2\,|D|\,\ell_V$, at which point the death rate is dropped back to 10%. So the population size $|S|$ is maintained around $1.2\,|D|\,\ell_V$ by a negative feedback process, to ensure that the population size $|S|$ is always greater than $|D|\,\ell_V$.

Organisation
The Organisation class implements the Physical complexity for VLPs, C_V in (39), to determine the organisational complexity. The involves calculating ℓ_V and then the per-site entropies.

An alpha version of the Physical complexity for VLPs with epistatic interactions (which is detailed in Appendix C) is implemented. First the wild-type is determined and then used to generate mutational clones, similarly to the process described in section 4.1.3. The reason it is classified as an alpha implementation is because the definition in Appendix C is incomplete.

5.2 Hypotheses

The first test is not a hypothesis, but a requirement to ensure that the evolutionary computing process is working correctly. This will be done by analysing the average and maximum fitness of the population over the generations. The graphs should look as follows:

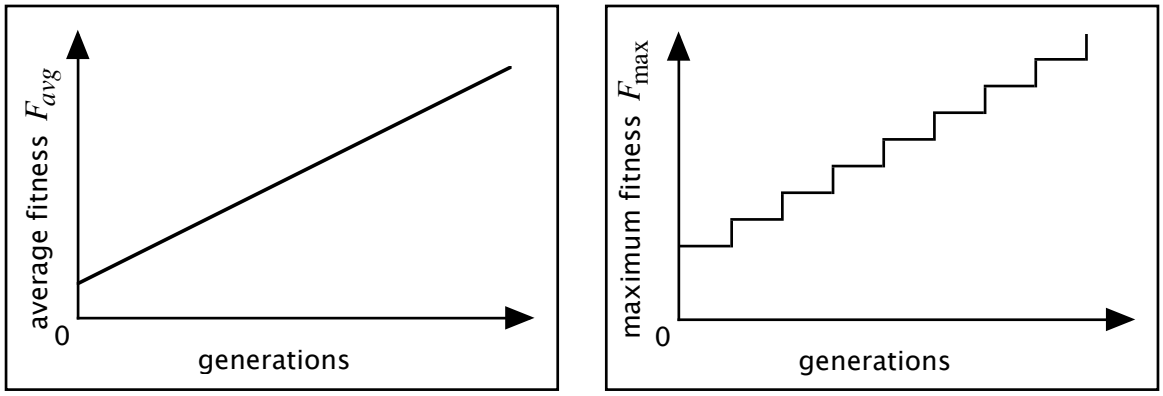


Figure 19: Expected Results for Average and Maximum Fitness

The average fitness is expected to begin small and increase steadily over the generations, always remaining below the maximum fitness. The maximum fitness is expected to begin small and increase steadily over time in jumps, because it is dependent on the arrival of new mutants (sequences) to increase.

After the evolutionary computing process is confirmed working, the primary concern will be to ensure that the modified measure (39) acts as expected, so that the later work upon which it is built is viable. To ensure a controlled test, the selection pressure will be static and alphabet(agent pool) will remain constant throughout, as specified in the section 5.1.3. Then, secondary experiments will be performed on the efficiency measure and clustering.

5.3 Visualisation of Population

Although not a hypothesis, a visualisation should be created to show the organisational complexity of the population visually, so that intuition can also be used in determining the results with respect to the hypotheses. The form of the visualisation should be similar to the visualisation of populations, as illustrated in the Figures of the earlier sections of this deliverable.

5.4 Physical Complexity for VLPs

The modified measure C_V in (39) is not only expected, but is required to increase over the generations, and suffer dips from the arrival of new mutants. The change to VLPs should not affect this, and if it does then the construction of the modified measure would be flawed.

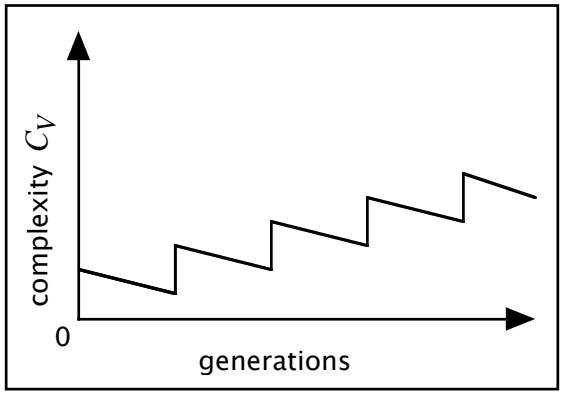


Figure 20: Expected Results for Physical Complexity for VLPs

The Physical complexity C_V can is expected to increase over the generations, as the complexity potential will increase as the length of the agent-chains increase. Realised complexity (information) increases with the uniformity of sites across the population, which will in general increase over the generations. However diversity, such as the arrival of a new fitter mutation can cause the Physical complexity C_V to decrease in the short-term, due to the temporary increase in diversity.

5.5 Efficiency Measure

The efficiency measure should converge towards its maximum one, in populations with only one cluster. This convergence should be indicated by the fitness measures, specifically the average fitness converging to the maximum fitness.

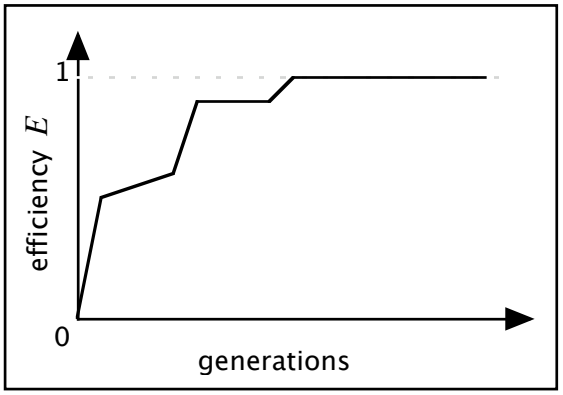


Figure 21: Expected Results for Efficiency

The efficiency is expected to tend towards one, although erratically, as it is dependent on the complexity C_V which itself suffers from short-term falls.

5.6 Investigate Clustering

The efficiency measure E (clustering coefficient) should act in conjunction with the fitness measures (clustering indicator) to allow the determination of the number of clusters. The agents in the AgentStation will be instantiated to allow the formation of two pure clusters

within the population. They will be measured after the population reaches equilibrium. So the efficiency E, due to the multiple clusters within the population, should tend towards a value significantly below the maximum:

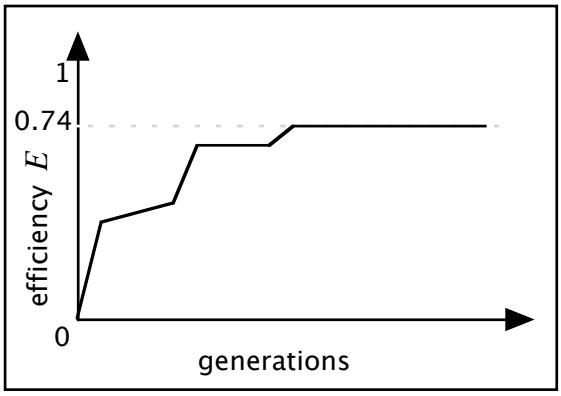


Figure 22: Expected Results for Efficiency with Two Pure Clusters

The efficiency E should tend towards 0.744, calculated from (47):

$$\begin{aligned} \textit{given}: \quad & |D| = 15\,|T| = 2(47) \\ \textit{hence}: \quad & E \to 1 - (\log_{|D|}|T|) \\ & E \to 1 - (\log_{15} 2) \\ & E \to 0.744 \end{aligned}$$

The efficiency E will tend towards 0.744 rather than the maximum, as there will be two clusters within the population. This will reduce the uniformity across the sites, and therefore the complexity C_V (39) upon which the efficiency E depends.

The efficiencies E of the two clusters should both be near their maximum, which is one (43).

6 Results

The graph below shows the maximum and average fitness over the generations. These basic results from the simulation match the requirements in section 5.2, Figure 19, showing that the evolutionary process is working.

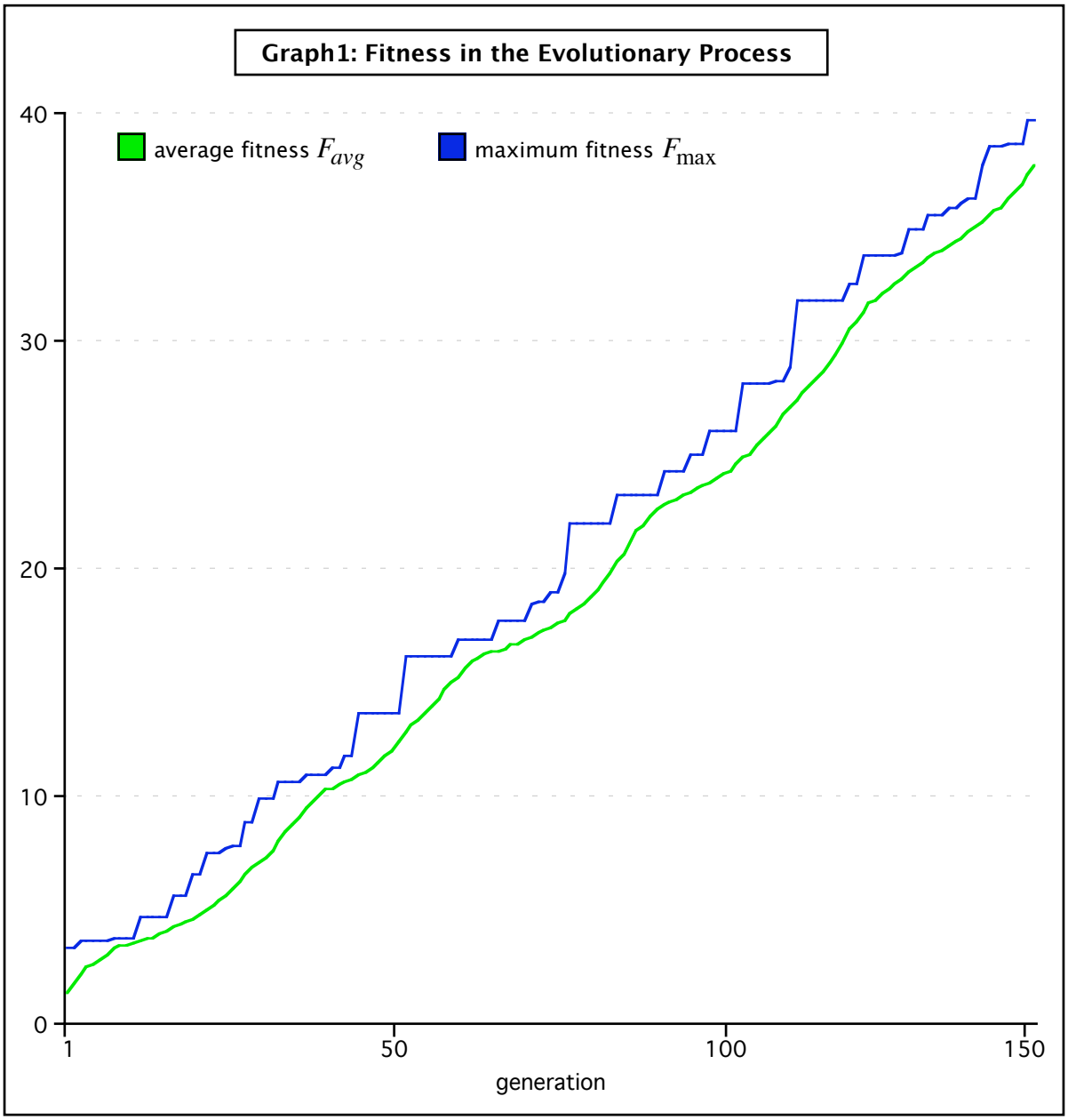


Figure 23: Graph 1: Fitness in the Evolutionary Process

The graph clearly shows the maximum fitness only ever increasing over the generations. The average fitness also increases over the generations, and as expected remains below the maximum fitness due to the variation in the population.

It is clear that the evolutionary computing process is working satisfactorily for the complexity measure to be investigated.

6.1 Visualisation of Population

A visualisation was created with the simulation in conjunction with a spreadsheet.

The visualisations presented below shows the extremes of organisation and disorganisation, within an evolving agent population. Both populations were run for a 1000 generations, the one on the left under normal conditions, and the one on the right without a selection pressure. Each coloured square  represents and individual agent, and a horizontal line of coloured squares  represents an agent-chain.

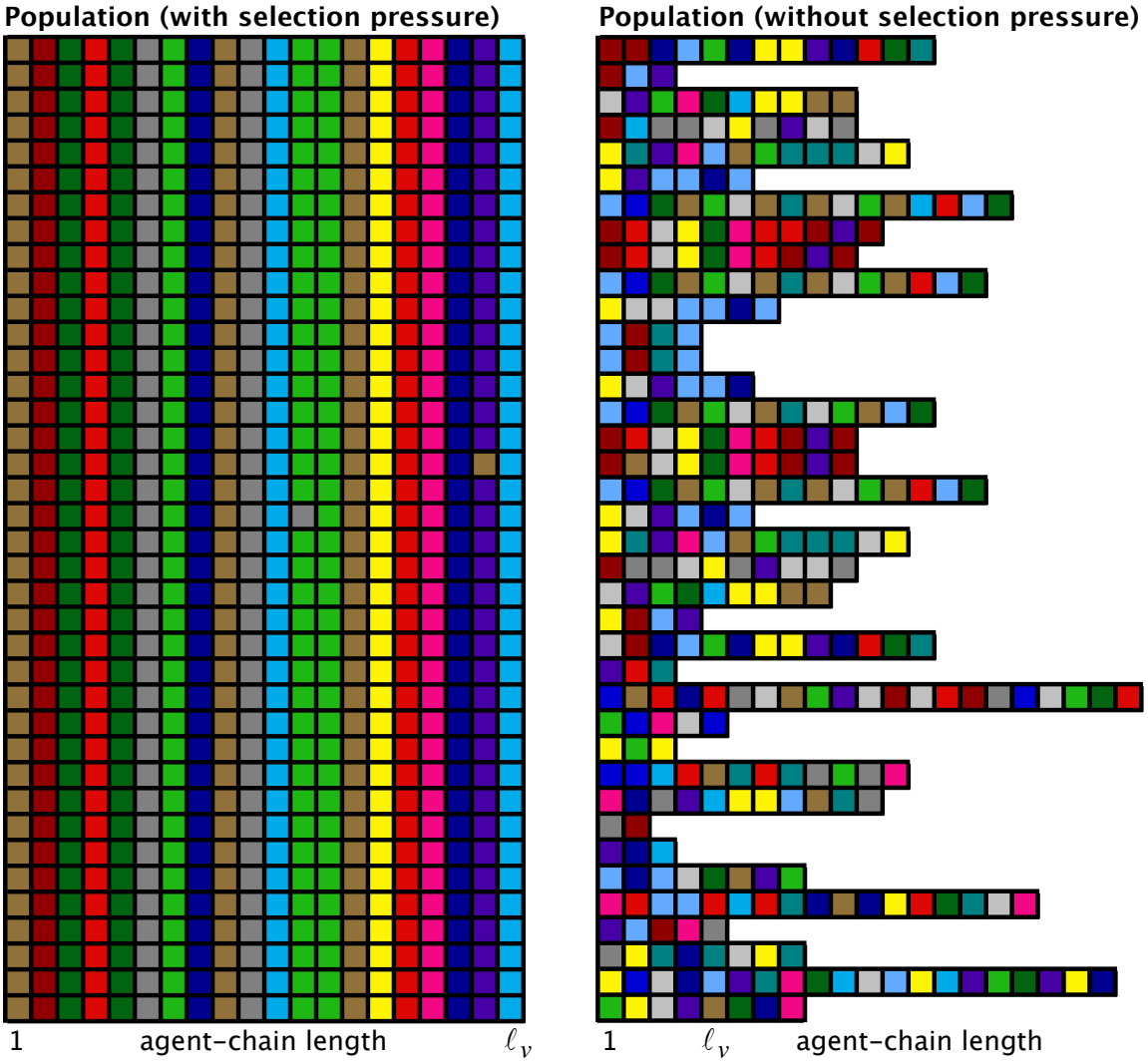


Figure 24: Visualisation of Organisational Complexity

This view only shows about a tenth of each population. This sample is shown at this scale, so that the individual agents and agent-chains can be seen. The difference in organisational complexity is obvious. The population on the left has two squares out of place in the visualisation. This is expected and is due to mutation.

Compressed representations of both populations in full are shown below.

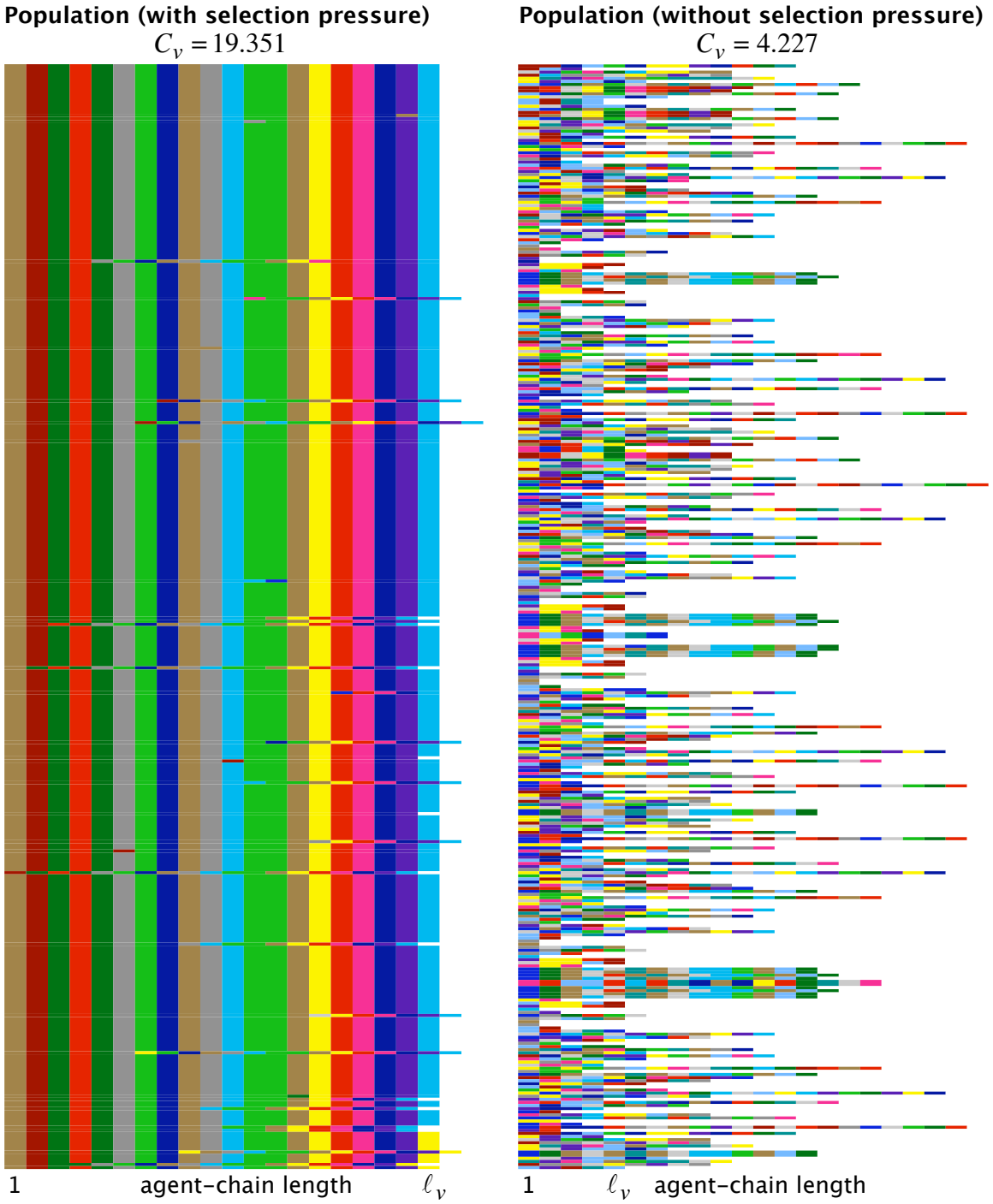


Figure 25: Visualisation of Organisational Complexity 2

Viewing the full populations the difference is even more evident, and shows that their respective complexity C_V (39) values are accurate in describing their organisational complexity.

The visualisation also shows, that the highly ordered population on the left is not without significant variation. This is good, because it shows that the evolutionary computing process creates the opportunity to find fitter(better) solutions.

6.2 Physical Complexity for VLPs

Below shows a graph of the Physical complexity modified for VLPs C_V (39), against the generation. Although the complexity with epistasis measure C_{V_e} has not been fully formalised (see Appendix C for partial definition), as it ultimately relies on population sampling a prototype version has been implemented. The maximum fitness F_{max} has also been included.

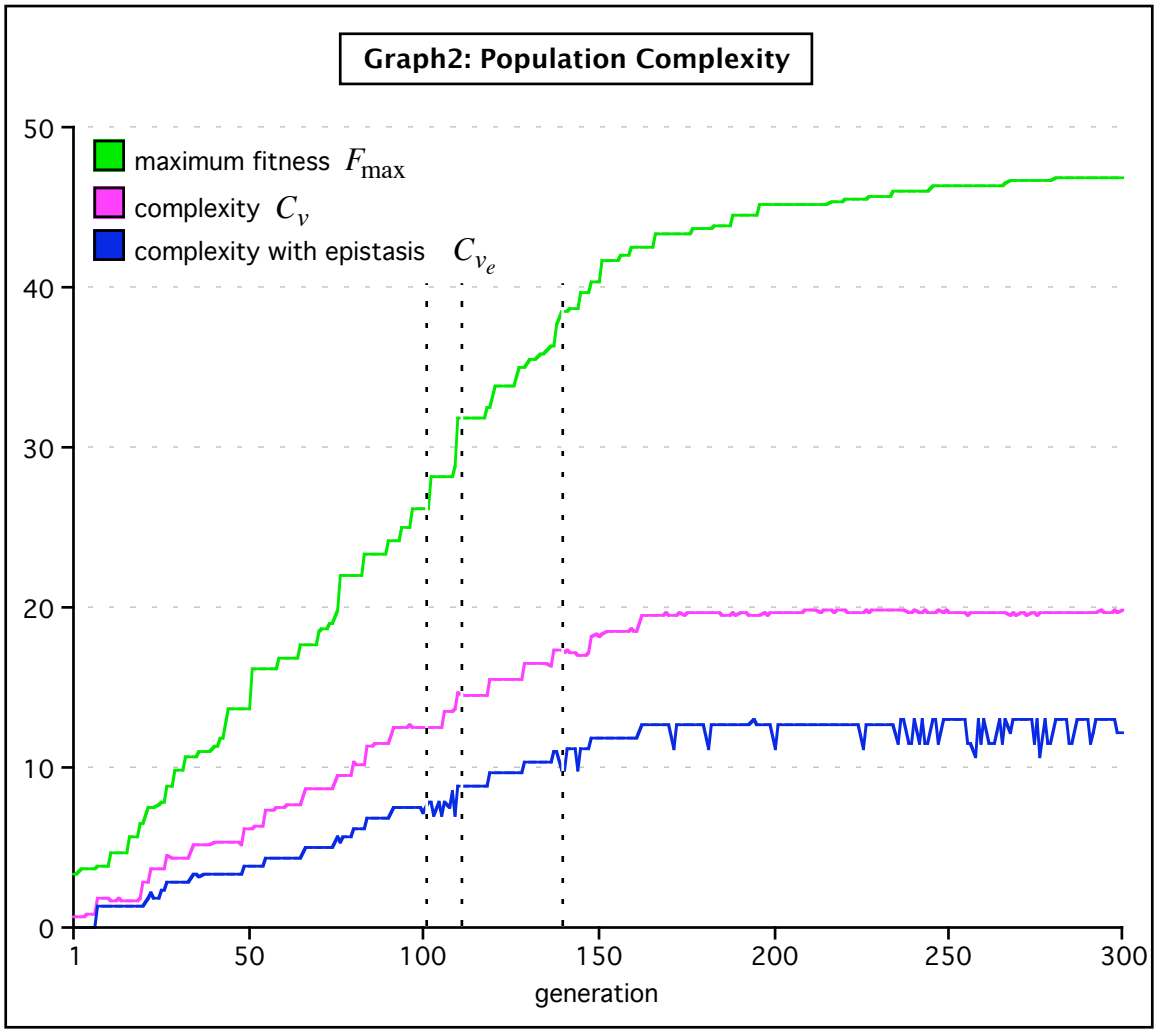


Figure 26: Graph 2: Population Complexity

The complexity for a VLP, C_V, increases over the generations, and shows short-term falls as expected. The increase is due to increasing information being stored, and the small jumps is when the effective length ℓ_V of the population increases. A fall that starts at generation 140 is preceded by the arrival of a new fitter mutant, shown by a jump in maximum fitness F_{max}.

The complexity for a VLP with epistasis C_{V_e} in (C.7) also increases over the generations, but at a lower rate indicating increasing epistasis. The sporadic measurements at generation 100 to 110, and throughout, are due to the difficulty in choosing appropriate wild-types, which will be discussed later.

6.3 Efficiency Measure

The graph below shows the efficiency measure $E(40)$ over the generations. The efficiency measure $E(40)$ is the complexity $C_V(39)$ over the complexity potential $C_{V_e}(32)$.

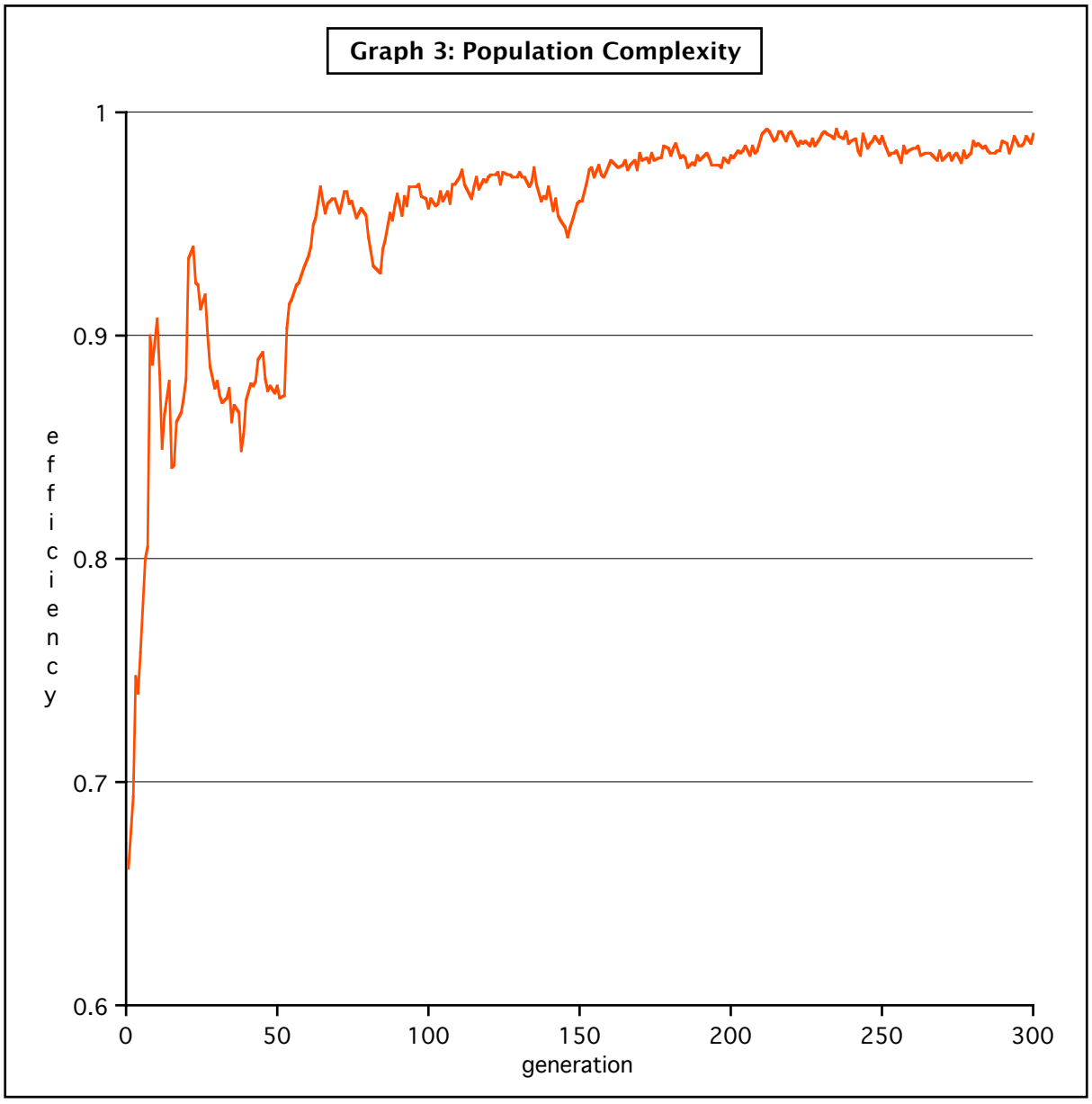


Figure 27: Graph 3: Population Efficiency

The population being used has only one cluster, so the efficiency $E(40)$ tends to its maximum 1, as expected. The significant falls on the way towards the maximum, can be seen to decrease in magnitude and frequency over the generations. The measure is directly dependent on the complexity C_V, so falls in C_V are mirrored here. These falls are caused by the creation of fitter(better) mutants within the population, which eventually become the dominant genotype, but during the process cause complexity and efficiency to fall in the short-term.

6.4 Investigate Clustering

The scenario considered, is a population with two pure clusters, such that the clusters share no agents. The clustering indicator, average fitness tending towards maximum fitness, $F_{avg} \rightarrow F_{max}$, is shown in the graph below.

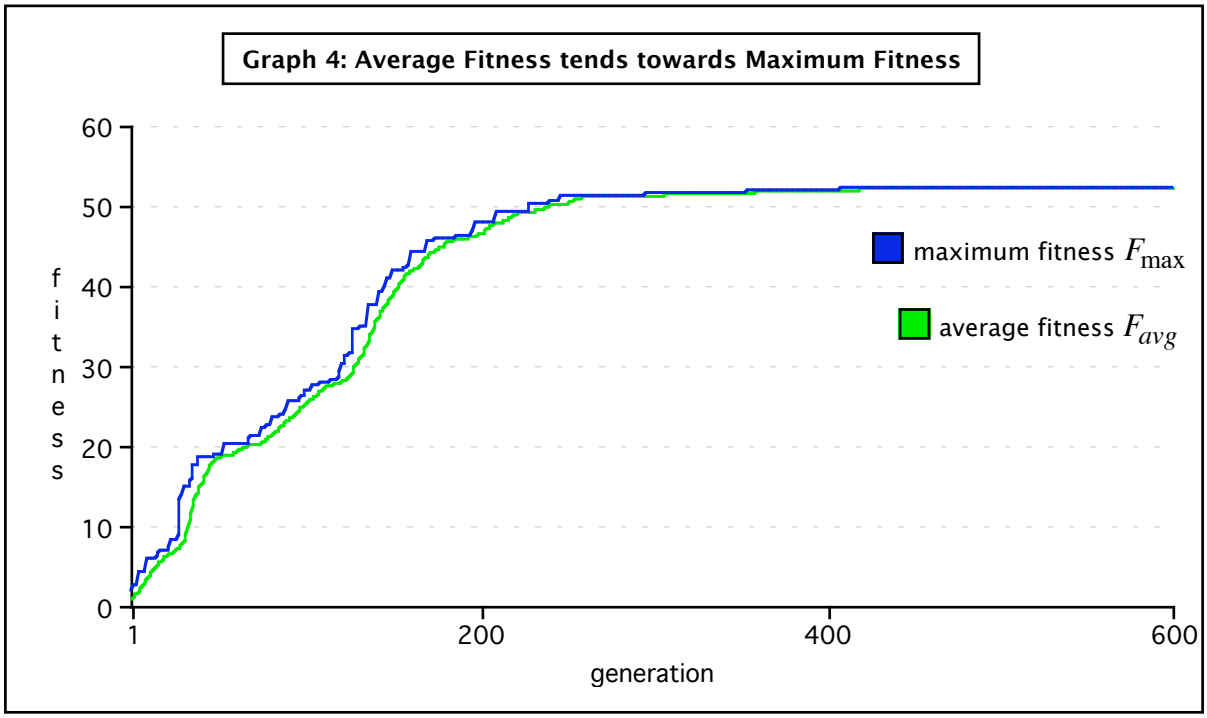


Figure 28: Graph 4: Clustering Indicator

The average fitness tends towards the maximum fitness indicating clustering.

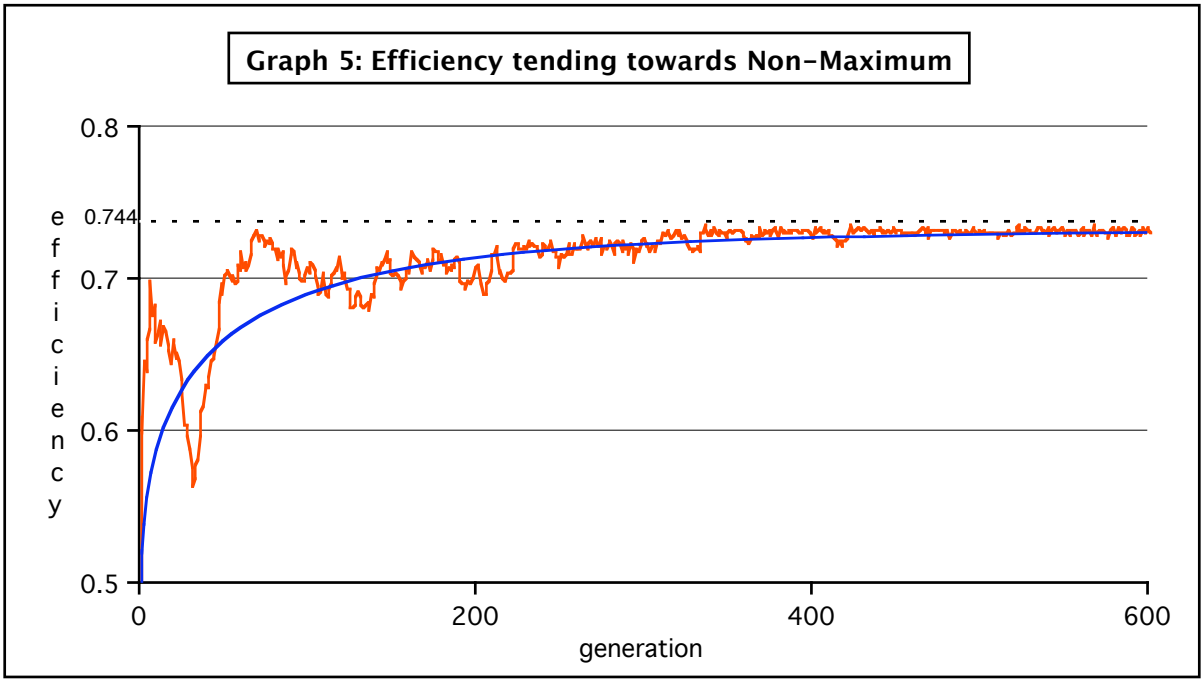


Figure 29: Graph 5: Clustering Coefficient

The efficiency E, the clustering coefficient, tends towards a value significantly below the maximum of one, 0.744 as predicted. This indicates clustering with more than one cluster in the population. A visualisation of the population is shown below, indicating the clusters.

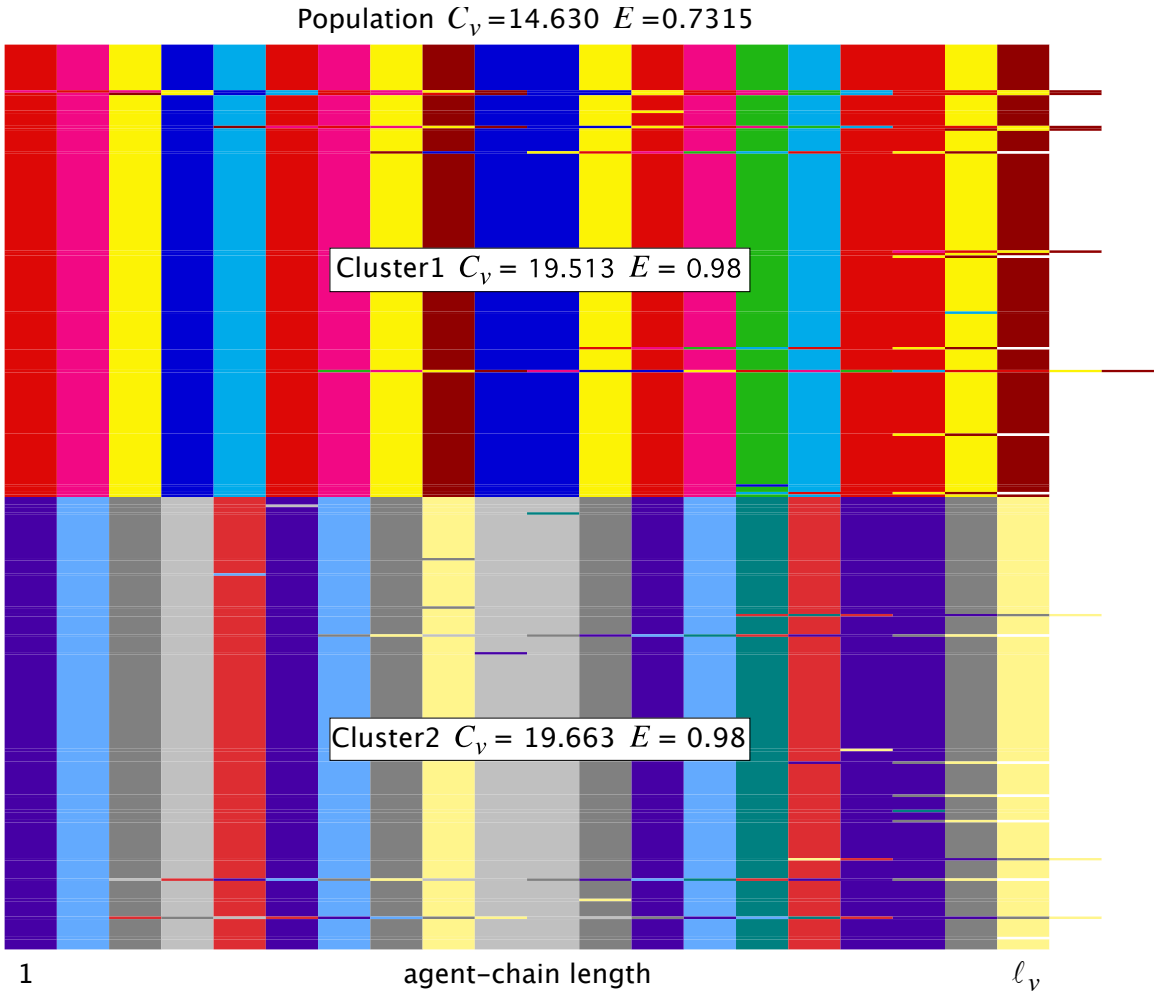


Figure 30: Visualisation of Clusters

The population contains two clusters, which can be clearly seen from the visualisation. The clusters, as expected, each have a much higher complexity C_V on their own, and near maximum efficiency E on their own.

The efficiency E shown in Graph 5 (Figure 29) appears to oscillate around the best fit curve until it comes to sit on top of it, only showing minor oscillations away from it. The large peaks and falls initially are due to the creation of fitter(better) longer mutants(agent-chains) in the population. As they spread initially they cause falls in efficiency, but once they take over the population the efficiency rises to a new higher level.

7 Conclusions

First consideration will be given to the achievements in the modelling process and the experimental results. Then, the limitations and how they can be managed will be considered.

7.1 Achievements

The achievements, their significance, and implications with respect to both the modelling and the simulation are discussed.

7.1.1 Abstract Model

An abstract model for the DBE service composition in the Evolutionary Environment(EvE) has been constructed in terms of a Multi-Agent System(MAS). It would be more accurate to say that the Evolutionary Environment was defined from the MAS model, as that is what actually happened. I came up with the interconnected habitats based on the connectivity of agents stations in a MAS. So, provided that the EvE is implemented faithfully, there should be little concern of the appropriateness and accuracy of the MAS model for the service composition within the EvE. For the same reason, the simulation constructed provides viable experimental results, as the EvE model and its preliminary class diagrams were constructed from the prototypes of the simulation. Finally, the flexibility of the MAS model, due to the nature of software agents, will allow the work to continue without suffering from any technology changes that may occur in the DBE service implementation. Agent as a synonym for DBE service might be subject to future change.

7.1.2 Definition for Organisational Complexity

The alternative measures for representing organisational complexity in the DBE services have been evaluated and considered for their suitability to the target system. The Physical complexity measure was then chosen for further investigation, as its properties closely matched our intuitive definition of organisational complexity.

The Physical complexity measure has been studied in detail, and a review of the literature beyond that of the measure's author, Adami, has been performed to confirm that the measure is the most suitable choice. It effectively measures a populations organisational complexity relative to the environment, without actually considering the environment (selection pressure/fitness function) directly. This is a very powerful and useful property which effectively insulates the formulation of the measure from any changes in the fitness function.

It has been adapted to provide a general definition of organisational complexity (organisation) in agent populations of real multi-agent systems, rather than the near machine code self-replicating programs of the Avida artificial life simulator[2]. This definition of Physical complexity matches the intuitive definition of organisational

complexity in a population. Most significantly it has been reformulated algebraically, for populations of variable length individuals, which has been shown to be correct experimentally through simulations.

7.1.3 Performance Measure

An effective performance measure, the efficiency E (40), has been constructed from the definition of Physical complexity for variable length populations, and describes the efficiency of information storage within a population.

The efficiency measure is always between zero and one, so it can be used to compare any two populations, independent of their size, their length, and whether the length is variable or not.

7.1.4 Clustering

The Physical complexity measure intrinsically and when extended to variable length populations, has minimal requirements to be applicable. This means it can cater for many different scenarios. One being clustering, for which the preliminary work is promising.

The clustering coefficient (45) defined by the tending of the efficiency E (performance measure), not only indicates the level of clustering, but can also distinguish between a population with a single cluster and population of multiple clusters. In the case of pure clusters, the number of clusters can be determined from the value that the clustering coefficient tends towards.

7.1.5 Atomicity

This atomicity of agents, requiring that no single agent can functionally replace an agent-chain of length two or more, represented a substantial challenge. It would have been an unrealistic constraint to propagate form the model to the real-world. Non-atomic agents were problematic, because they affect the summing of the per-site entropies, which is the main construct of the Physical complexity measure.

The non-atomicity in itself causes clustering, so the clustering concepts were used to resolve the problem by creating a variant of the efficiency measure for multiple clusters $E_m(50)$. Thereby removing the effect on the per-site entropies, as the complexity of the clusters are calculated separately, and their individual efficiency measures averaged. So, atomicity is no longer required for the application of the Physical complexity measure.

7.1.6 Experimental Results

The experimental results though preliminary, are very promising.

Firstly, the evolutionary computing dynamics are sufficient to allow investigation of evolving agent populations, shown by the similarity of Graph 1 (Figure 23) to its

predictions in Figure 19. The only difference is that the line of the average fitness was not straight as predicted, which comes from normal fluctuations in the evolutionary dynamics.

The results show that the Physical complexity measure of organisational complexity has been successfully extended for variable length populations, shown by the similarity of Graph 2 (Figure 26) to its prediction in Figure 20. However, the falls in the Physical complexity C_V were not always as severe as predicted. The reason for falls is the arrival of new mutants to the population. The falls were small when only a single mutation had generated the mutant, whereas large falls occurred when the mutant had several mutations. Thus, the fundamental reason is because the mutation rate was relatively low.

The results for the efficiency E (performance measure) were as predicted, shown by the similarity of Graph 3 (Figure 27) to its prediction in Figure 21. Although the presumed oscillations were underestimated, and were caused by the increasing length ℓ_V in the population. As ℓ_V increases, so does the complexity potential C_{V_P}, which is the denominator of the efficiency E. So as the population length ℓ_V came to stabilise to its optimum, so did the efficiency E.

With respect to the clustering, the clustering indicator, average fitness tending towards maximum fitness, converged as predicted, as shown by Graph 4 (Figure 28). The clustering coefficient, efficiency E, under clustering tended towards the predicted value, shown by Graph 5 (Figure 29) compared to its prediction in Figure 22. Again, here the oscillations were underestimated, and have the same cause, the initially increasing length ℓ_V.

Overall, the simulation results have supported the hypotheses, and have provided more detail to the behaviour of the phenomena under investigation.

7.1.7 Visualisation

The results can be visualised to allow results to be checked intuitively as well as numerically. It is an important tool for comparing the intuitive understanding from the visualisation to the numerical conclusions.

7.2 Connection to Other Work Packages

This work is preliminary, but once finished will be applicable to the Evolutionary Environment[10] which is currently being integrated into the DBE Core Architecture[17]. The efficiency, as a performance measure and clustering coefficient, is useful as an analytical tool for populations in the Evolutionary Environment which is specified in the Evolutionary Environment Discussion Paper[10], and is a product of C42 Design of Intelligent System/Distributed Storage.

7.3 Limitations

None of the limitations are so severe, that they cannot be resolved or managed, but they of course have potential implications which must be considered.

7.3.1 Possible Limitations of the Chain Structure

The one significant limitation of the Physical complexity measure is the requirement that when agents/services aggregate, they do so in the form of chains. Their linear structure is the basis for the measure. All current work within the DBE is using the agent-chain as a basis. There have been discussions suggesting that the aggregation of DBE services in the opportunity spaces (business sectors) under consideration, may not use or require the chain structure. Specifically, for the manufacturing industry, supply chains, to which DBE service-chains are often compared, can take the form of tree structures, but in their simplest form are equivalent to the linear chain structure proposed. Also, a supply tree can be split into two chains with partial duplication. The tourism opportunity space need not have the DBE services chained. Consider the processes in a package holiday; flight, hotel, limousine. They need not be organised into a chain, as they do not have strong interdependencies, for example airlines do not refuse to serve you based on the hotel that you use. Although the tourism scenarios do not require chains, they can be represented by chains, based on the expected order in which events are supposed to occur, i.e. you want a limousine after the flight lands.

Based on the currently available information for both, yet to be fully defined opportunity spaces, the chain structure can be fully applicable to both the manufacturing sector and the tourism sector. New developments or information may change this, but at the moment on the available information the chain structure is a viable assumption and with respect to organisational complexity the chain structure is the most logical step from the atomic units (services).

7.3.2 Sporadic Results in the Epistatically Capable Complexity Measure

The sporadic results for the variant of the Physical complexity measure for managing epistatic effects (C_{V_e} in Figure 26, Graph 2), could well be due to the difficulty in finding a wild-type(agent-chain) which is not epistatically linked, and, or, an implementation error caused by the definition not being finalised for variable length populations, to which it was applied (see Appendix C for the partial definition). It was perhaps somewhat ambitious to attempt an application of an incompletely defined measure, as the unexpected sporadic behaviour cannot be satisfactorily explained.

The value of this variant of Physical complexity depends on a case by case basis on the amount of epistasis in the population. The amount of epistasis is primarily determined by the interdependencies caused by the fitness function which is defined from the user request. Therefore, if the epistasis is significant, it is worth using the measure else it provides no additional information. If it were to be found that the epistasis is similar in all requests, then for inter-population comparison the original variant of the Physical complexity measure would be sufficient.

7.3.3 Clustering Indicator

The clustering indicator, although not a construct of the complexity measure, but of the fitness information is very useful for indicating if the efficiency measure is tending towards

its limit. When the efficiency tends towards its limit, the exact numerical value is not known, and the plot of the measure oscillates uncertainly. Plotting the *log* of one or both of the variables unfortunately yields no better indication. Therefore, the fitness based clustering indicator is very useful in indicating that clustering is occurring, and then the efficiency measure can be used to determine the number of clusters.

7.3.4 Clustering

In the results, only the visualisation of clustering in the population after the evolutionary process had reached equilibrium, and the original efficiency measure E could be provided. The efficiency E_m for multiple cluster previously defined could not be implemented efficiently, because the clustering, as currently defined algebraically, lacks a computationally feasible definition for finding the clusters. The solution is not obvious, although counting the genotype frequencies with some kind of fuzzy matching could provide the basis of a solution, because the alternative of evaluating the complexity many times over, could be computationally expensive.

7.3.5 Mutation

Only point mutations have been considered in the evolutionary system. The technique of crossover, where two agent-chains are crossed at a point in their chains and the later half of each exchanged, it not included. This or another form of recombination which would help to minimise epistatic effects in the population.

7.4 Summary

We recommend adopting the Physical complexity measure for VLPs as the organisational complexity (organisation) measure, based on the strength of the experimental results. The core work has been done in defining the organisational complexity (organisation), and in creating an effective performance measure from the Physical complexity, which has been tested experimentally with the simulations. With the Physical Complexity as the organisational complexity (organisation) measure and an abstract model of the DBE, we can begin to investigate deeper questions described in section 8.2. The simulation will be used for experimenting with different scenarios and algebraic formulations, so that the intuitive understating from the visualisations can be compared to the numerical results.

8 Future Work

This section describes some of the possible ways in which the existing work presented, can be extended in the future.

8.1 Extending Existing Work

This subsection describes how the existing work can and should be extended or finalised, so that the concepts presented have been fully investigated.

8.1.1 Extending Physical complexity with Epistasis for VLPs

The variant of the Physical complexity measure, to manage epistatic effects in VLPs, has not yet been fully defined (see Appendix C for the partial definition). Extending it to VLPs, has algebraically proved more difficult than expected due to the difference of point mutations in VLPs compared to same length populations. Point mutations now include deletions and insertions, as well as replacements upon which the original version is based. So, the task is to complete the definition and investigate it using the simulation.

8.1.2 Study Clustering Further

It is necessary to define what exactly clustering means, how much or how little sharing of sites between clusters is allowed in its definition. This will include consideration of its relationship to speciation (formation of new species) in evolution.

A computationally feasible method for determining the clusters in the population needs to be found, most likely using the genotype frequencies.

Consideration will also be given to how the clustering(efficiency) will work with multiple populations within a single habitat, and distributed populations in multiple habitats.

8.1.3 Multiple Optima Performance Measure

The efficiency E_m for managing populations with multiple clusters needs to be implemented and experimented with in the simulation. It should be a truly scalable measure of efficiency in the population, because it will be independent of the number of clusters, non-atomic agents and length (variable or fixed).

8.1.4 Changing Conditions: Migration & User Behaviour

Changing conditions includes additions and deletions to the alphabet of agents at the agent-station(habitat), caused by migration and extinction. Constantly changing user requests (selection pressure), which will require the simulation of user behaviour. These scenarios are all equivalent to evolving a solution to a moving target.

8.1.5 Tree Structure

Reformulation of the Physical complexity for tree structures, could prove to be a very challenging and fruitful task, and it may become a necessity if the DBE service structure is eventually defined as trees rather than chains.

8.2 New Ideas

This subsection describes entirely new ideas which may be based upon the existing work.

8.2.1 Co-evolution & Ecosystem

The total organisational complexity of an ecosystem could be defined as the mutual entropy of all organisms, about each other and the world in which they live. This is an information-theoretic formula that is not difficult to write down, but the associated quantity promises to be much more difficult to measure[2].

The simplest scenario of this would involve analysing two populations in the same habitat(agent-station) or in different ones, which share solutions with one another. The challenge will be determine how best to measure the combined organisation of separate populations.

8.2.2 Ecosystem Organisation

Can standard techniques for describing ecosystems be applied to the DBE. An energy pyramid is used to show the dissipation of energy, at each trophic level (producer and consumers). A food web shows the consumption dependencies between species in the ecosystem. Do equivalent structures exist for the DBE ?

8.2.3 Metapopulation Model

When we consider the DBE beyond the scope of a single population, therefore looking at the interconnected agent stations (interconnected Habitats of the EvE[10]), then the metapopulation model may useful. A metapopulation is a concept from biology, and is a collection of relatively isolated, spatially distributed, local populations bound together by occasional dispersal between populations[22].

The regional metapopulation persists in the face of local extinctions because of sufficient dispersal (service migration) among populations. If dispersal among populations is so frequent that local extinctions do not occur, then the regional population is better thought of as a single spatially distributed population. Such a scenario would most likely be avoided in the Evolutionary Environment. Nevertheless the metapopulation model should be investigated further for its applicability to the DBE.

8.2.4 Self-organising Systems to Dissipate Energy

Self-organising systems which dissipate energy (increasing rate of entropy) appear to match the behaviour of ecosystems. An example of a self-organising system that is well known for increasing the rate of entropy increase, is the vortex that forms when draining fluids through small holes.

Figure 31: Self-organisation - Increasing Rate of Entropy Production

The question is whether the DBE can be considered to be such a system. The problem has many facets, but the first would have to be whether energy is equivalent to CPU time, and matter is equivalent to information. Also, what is the mapping of money or transactions. Furthermore, this thermodynamical entropy is not the same as the entropy defined in the main text.

The interesting paradox is of the organisation in biological systems, not being fundamentally possible due to the second law of thermodynamics. This is always explained as local organisation at the expense of increased global disorganisation, but as a fix rather than a reason. The alternative of self-organising systems to dissipate energy, is that it would make self-organisation inevitable rather than unexpected. Let us not forget biological life, is the most efficient dissipator of stored energy, consider human use of fossil fuels.

8.2.5 Communication with Business Partners

When this work is completed, as stated in the Technical Annex[13], at month 18 deliverable D6.3 will provide an analogy of multi-agent systems and composed business services for communication with the business partners in the project.

References

[1] C Adami. Self-organized criticality in living systems. *Physical Review Letters*, A 203:23, 1995.

[2] C Adami. What is complexity? *BioEssays*, 24:1085–1094, 2002.

[3] C Adami. Sequence complexity in darwinian evolution. *Complexity*, 8(2):49–56, 2003.

[4] C Adami and N Cerf. Physical complexity of symbolic sequences. *Physica D*, 137:62–69, 2000.

[5] C Adami, C Ofria, and T Collier. Evolution of biological complexity. *Proceedings of the National Academy of Sciences*, 97(9):4463–4468, 2000.

[6] W Ashby. Principles of the self-organizing dynamic system. *Journal of General psychology*, 37:125–128, 1947.

[7] A Barron, J Rissanen, and B Yu. The minimum description length principle in coding and modeling. *IEEE Transactions on Information Theory*, 44:2743–2760, 1998.

[8] G Basharin. On a statistical estimate for the entropy of a sequence of independent random variables. *Theory Probability and its Applications*, 4:333–336, 1959.

[9] W Beck, K Liem, and G Simpson. *LIFE: An Introduction to Biology*. HarperCollins Publishers Inc., 3 edition, 1991.

[10] G Brisoce, J Rowe, and P Dini. Evolutionary environment discussion paper, 2004.

[11] G Chaitin. Algorithmic information and evolution. In *Perspectives on Biological Complexity*, pages 51–60. IUBS Press, 1991.

[12] A Corallo. Bml draft 2.2. 2004.

[13] P Dini. Annex I - "description of work". 2002. Available from: `https://dbe.digital-ecosystem.net/files/documents/8/6/file_6.dat?filename=DBE%20Annex%20I_ffinal.doc.gz`.

[14] R D'Souza and N Margolus. Thermodynamically reversible generalization of diffusion limited aggregation. *Physical Review E*, 60:264–274, 1999.

[15] S Elena and R Lenski. Test of synergistic interactions among deleterious mutations in bacteria. *Nature*, 390:395–398, 1997.

[16] J Epstein. Zones of cooperation in demographic prisoner's dilemma. *Complexity*, 4:36–48, 1998.

[17] P Ferronato. DBE core arch scoping document v1.0b, 2004. Available from: `https://dev.digital-ecosystem.net/files/documents/20/174/file_174.dat?filename=DBE%20Core%20Arch%20Scoping%20Documentv10b.pdf`.

[18] H Flyvbjerg, K Sneppen, and P Bak. Mean field theory for a simple model of evolution. *Physical Review Letters*, 71:4087–4090, 1993.

[19] C Gershenson and F Heylighen. When can we call a system self-organizing? In *Advances in Artificial Life, 7th European Conference, ECAL 2003 LNAI 2801*, pages 606–614. Springer-Verlag, 2003.

[20] N Gibbins, S Harris, and N Shadbolt. Agent-based semantic web services. *Journal of Web Semantics*, 1:141–151, 2004.

[21] H Gutowitz, J Victor, and B Knight. Local structure theory for cellular automata. *Physica D*, 28:18–48, 1987.

[22] I Hanski and D Simberloff. *The metapopulation approach, its history, conceptual domain, and application to conservation*. Academic Press, 1997.

[23] M Huhns. Agents as web services. *IEEE Internet Computing*, 2:93–95, 2002.

[24] A Kolmogorov. Three approaches to the definition of the concept "quantity of information". *Problems of Information Transmission*, 1:1–7, 1965.

[25] T Kurz. Dbe term glossary 04070 v00.00.01. 2004.

[26] R Lenski, C Ofria, T Collier, and C Adami. Genome complexity. *Nature*, 400:661–664, 1999.

[27] Z Maamar, Q Sheng, B Benatallah, and G Al-Khatibc. A three-level specification approach for an environment of software agents and web services. *Electronic Commerce Research and Applications*, 3:214–231, 2004.

[28] M Newman. Evidence for self-organized criticality in evolution. *Physica D*, 107:293–296, 1997.

[29] G Nicolis and I Prigogine. *Self-organization in Nonequilibrium Systems: From Dissipative Structures to Order Through Fluctuations*. John Wiley & Sons Inc, 1997.

[30] H Parunak and S Brueckner. Entropy and self-organization in multi-agent systems. In *Proceedings of the Fifth International Conference on Autonomous Agents*, pages 124 – 130, 2001.

[31] A Prügel-Bennett. Modelling evolving populations. *Journal of Theoretical Biology*, 185:81–95, 1997.

[32] A Redmore and M Griffen. *Longman Reference Guides: Biology*. Longman Group Limited, 7th edition, 1994.

[33] M Ridley. *The Cooperative Gene*. The Free Press, Simon & Schuster Inc., 2001.

[34] J Rissanen. Lectures on statistical modeling theory. Technical report, Department of Computer Science, University of Helsinki, 2004. Available from: `http://www.cs.helsinki.fi/u/tmononen/SMT/rissanen_lect.ps` [cited 25/02/04].

[35] T Schneider. Information theory provider. Available from: `http://www.lecb.ncifcrf.gov/~toms/paper/primer/` [cited 14/06/04].

[36] C Shalizi and K Shalizi. Quantifying self-organization in cyclic cellular automata. In *Noise in Complex Systems and Stochastic Dynamics, Proceedings of SPIE*, volume 5114, pages 108–117, 2003.

[37] J Shapiro, M Rattray, and A Prügel-Bennett. The statistical mechanics theory of genetic algorithm dynamics. In *Proceedings of the First International Conference on Evolutionary Computation and Its Applications*. Kluwer, 1996.

[38] B Sleeper and B Robins. Defining Web Services. Available from: `http://www.perfectxml.com/Xanalysis/TSG/TSG_DefiningWebServices.pdf` [cited 15/04/04].

[39] K Sneppen, P Bak, H Flyvbjerg, and M Jensen. Evolution as a self-organized critical phenomenon. *Proceedings of the National Academy of Sciences of the United States of America*, 92:5209–5213, 1995.

[40] J Vidal. Learning in multiagent systems. Technical report, Swearingen Engineering Center, University of South Carolina, 2003. Available from: `http://jmvidal.cse.sc.edu/papers/vidal03a.pdf` [cited 21/04/04].

[41] W3C. Web Services Activity, 2004. Available from: `http://www.w3.org/2002/ws/` [cited 11/03/04].

[42] M Wooldridge. *Introduction to MultiAgent Systems*. John Wiley & Sons Inc, 2002.

A Interdisciplinary Dictionary

The key scientific terms used within this document will be explained here, in general terms and in terms of the Multi-Agents Systems(MAS) abstract model. These terms when first mentioned in the main text, will be defined there and then, relative to their context. These definitions in context, and this dictionary should hopefully make the terms clearly understood.

The scientific definitions are given in green italics[32, 9], and the MAS definitions are given in red. It is hoped that this dictionary will complement, and can be added to the other efforts to create a cross disciplinary glossary of terms within the DBE[25].

DNA
DeoxyriboNucleic Acid whose sequence encodes the genetic information of living organisms, provides two primary functions. The holder of virtually all information in inheritance, and the controller of protein synthesis.
The equivalent is the process descriptions associated with each agent.

ecosystem
An ecosystem is a natural unit made up of living and non-living components whose interactions give rise to a stable, self-perpetuating system. It is made up of one of more communities of organisms, consisting of populations existing in their microhabitats.

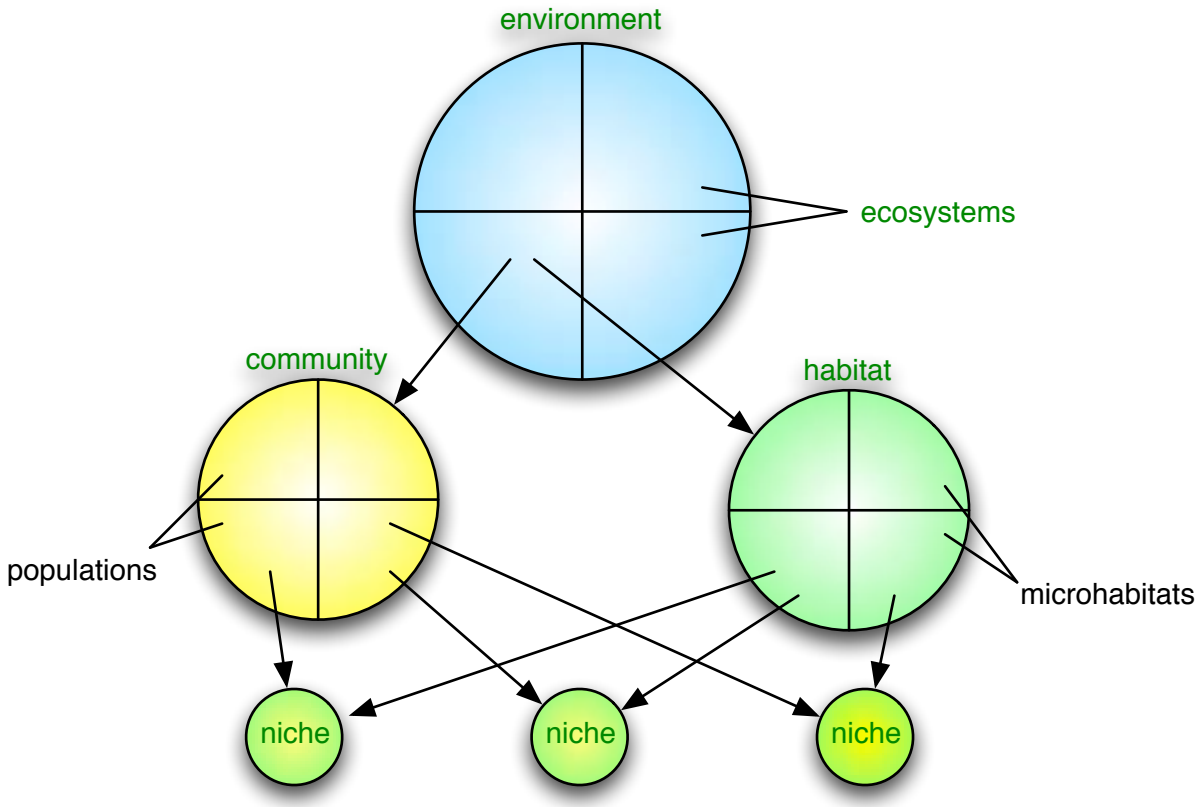


Figure A.1: Ecosystem

The agent-stations are equivalent to the habitats in Figure A.1, and the population objects are equivalent to the populations in Figure A.1. The fitness function is equivalent to the microhabitats in Figure A.1. The evolving population, under the influence of the fitness function, is equivalent to the niche in Figure A.1.

epistasis
An interaction between genes in which one gene masks or modifies the effects of another gene.
This is equivalent to the probability of finding a specific agent at a specific site in an agent-chain, being affected by the presence of other agents at other sites within the agent-chain.

evolutionary stable strategies (ESS)
It is based on the concept of a population of organisms, playing a certain strategy, and the mutant form of a gene that causes organisms to adopt a different strategy cannot invade the population, but will instead be selected out by natural selection.
This is equivalent to a population of agent-chains that have found all the global optima in the search space (fitness landscape). So any mutant agent-chains invading the population will be sub-optimal, and therefore removed by the selection pressure (fitness function).

fitness
Measure of the ability to produce mature offspring in the next generation which themselves will be able to reproduce.
This is equivalent to the fitness function used in the evolutionary computing, where fitness is dependent on the desired ability of the software requested.

gene
A small length of the total DNA sequence of an individual, to which a specific function can be assigned.
The equivalent unit is the process description of an agent.

gene pool
All the genes in a population or species, therefore its genetic constitution.
This is the set of agents registered at an agent station.

genotype
The genetic makeup of an organism, independent of it physical or functional traits.
This is equivalent to the process descriptions of agents and agent-chains.

migration
This is the movement of individuals between different habitats (areas), often due to seasonal variations.
This is equivalent to the movement of mobile agents from one agent-station to another.

mutation
A permanent transmissible change in genetic material(DNA).
In its simplest form, this is equivalent to replacing one of the agents in an agent-chain.

organism
An organism(phenotype) is an individual that is capable of reproducing itself and existing as a separate entity.
This is equivalent to the instantiation of an agent or agent-chain, which is generated by evolutionary computing, in response to a user request.

population
All members of a species that occupy a particular area at a given time. Statistically, the group of entities under study which can be sampled.
This is the population of agent-chains created, using Evolutionary Computing at an agent station, to generate a solution to a user request.

selection pressure/natural selection
Describes the sum total of the forces acting upon a population, resulting in genetic change and natural selection. Those organisms best fitted to survive the selection pressures operating upon them will pass on their biological fitness to their progeny through the inheritance process.
This is equivalent to the fitness function in the MAS model, which assigns fitness to the agent-chains of the evolving population, and thereby determines which will survive.

species
A series of populations within which significant gene flow occurs, so groups of organisms showing a very similar genetic makeup.
Pure clusters are an example of different species within the MAS model, as no agents are exchanged (no gene flow occurs) between them.

wild-type
The most frequent genotype within a population.
This is equivalent to the most frequent agent-chain within the population, which will potentially be instantiated at an agent-station in response to a user request.

B Spreadsheet Tool

Here you can investigate the Physical complexity measure by changing the alphabet size and population in red. Please note that it is only for fixed length populations.

Population S

1	2	3	4	
A	A	A	A	
A	B	B	C	
A	C	C	D	

	Persite Entropy		**Persite Information**
H(1)=	0.000	I(1)=	1.000
H(2)=	0.750	I(2)=	0.250
H(3)=	0.750	I(3)=	0.250
H(4)=		I(4)=	

alphabet |A,B,C,D

alphabet size |D|= 4 where: 1<=|D|<=5

population size |S|= 4 where: 1<=|S|<=5
length l = 3 where: 1<=l<=4

Complexity Potential Cp = 3

Complexity C = 1.500

Efficiency E = 0.500

%Efficiency %E = 50.00%

The spreadsheet can be found at:
http://dbe.hopto.org/PhysicalComplexity.xls

C Incomplete Definition

Epistasis in Physical Complexity for Variable Length Populations (VLPs)
This is also a significant part of the reformulation, because it is not an extension of the existing measure, but requires changing and re-justifying the fundamental assumptions.

To reformulate and re-justify the measure, it is necessary to understand its conditions and limits. The Physical Complexity, taking into account the epistatic interactions C_e, is the populations maximum entropy (length ℓ minus the population entropy $H(S|e)$:

$$in\ (24)\ we\ found\ tha\ tC_e = \ell - H(S|e)$$

Again the length ℓ, which no longer exists in a VLP, represents the maximum complexity, the complexity potential. The length for a VLP ℓ_V, as defined in (32), represents the complexity potential in a VLP. So ℓ_V can be substituted for ℓ. Therefore the Physical complexity for a VLP that takes into account epistatic interactions, C_{V_e}, is defined as:

$$C_{V_e} = \ell_V - H(S|e) \tag{C.1}$$

$H(S|e)$ remains as defined in (25), dependent on S_{viable}, which is the set of all viable sequences in the environment e, where viable means that the sequence can exist in the environment.

$$in\ (25)\ we\ found\ that\ H(S|e) = \log_{|D|} |S_{viable}|$$

In VLPs the set of all viable sequences, S_{viable}, could theoretically be an infinite set, as the number of possible genotypes is infinite, because there is no restriction on the length. However, in biology and in evolutionary computing, excessively long sequences are non-viable. In biology, excessively long sequences are said to suffer from mutational meltdown[33], and in evolutionary computing it is known as the problem of bloat.

To determine $|S_{viable}|$, the selection of a wild-type genotype as specified previously is still valid, with the additional constraint that the wild-type has length ℓ_V (length for VLP). However, the use of the wild-type in estimating the number of viable genotypes $|S_{viable}|$ changes, because the number of mutants generated at a mutational distance is not the same as before. The concept of mutational distance is the same, but the definition of a mutation is different.

For the original application of this measure, the mutation being applied to the digital organisms[5] was a point mutation, which simply replaced an instruction at the site being mutated. This was done because the population was of a fixed length, so the point mutation could not be allowed to change the length of the sequences.

The mutation, in an evolving population of agent-chains, can change the length, and is necessary to allow agent-chains to grow. So the definition of point mutations, includes replacements as before, and also insertions and deletions. So the definition of $w(n)$ in (27) must be modified to $w_V(n)$, to continue the derivation of determining the complexity C_{V_e}.

The form of $w_V(n)$ is currently not known, as it is a non-trivial matter to determine. However, $w_V(\ell_V)$ can be defined tentatively, as the number viable genotypes(sequences) at mutational distance ℓ_V, $|S_{viable}(\ell_V)|$, over the total number of sequences at mutational distance ℓ_V. Firstly, just as before $|S_{viable}(\ell_V)| = |S_{viable}|$. So (25) can be written as:

$$H(S|e) = \log_{|D|} |S_{viable}(\ell_V)| \tag{C.2}$$

Secondly, the denominator is the sum of all sequences at every length up to ℓ_V. The number of sequences at length ℓ_V, is $|D|^{\ell_V}$, the summation up to ℓ_V is:

$$\sum_{x=1}^{\ell_V} |D|^x = |D|^1 + |D|^2 ... + |D|^{\ell_V - 1} + |D|^{\ell_V} \tag{C.3}$$

As both $|D|$ and ℓ_V are positive integers, and this is a divergent finite geometric series, it can be estimated by the last value in the summation, because the last value is by far the largest in the summation:

$$\begin{aligned} \sum_{x=1}^{\ell_V} |D|^x &= \frac{|D|^{\ell_V+1} - |D|}{|D| - 1} \approx \frac{|D|^{\ell_V+1}}{|D|} \\ &= |D|^{\ell_V} \end{aligned} \tag{C.4}$$

So, $w_V(\ell_V)$ can be defined as the number viable genotypes at mutational distance ℓ_V, $|S_{viable}(\ell_V)|$, over the estimate of the total number of sequences at mutational distance ℓ_V (C.4).

$$w_V(\ell_V) = \frac{|S_{viable}(\ell_V)|}{D^{\ell_V}} \tag{C.5}$$

Previously the function of $w(n)$ was estimated in (28) as:

$$in\ (28)\ we\ found\ tha\ tw(n) = |D|^{-\alpha n^{\beta}}$$

Assuming that $w_V(n)$ can be estimated with the same function:

$$w_V(n) = |D|^{-\alpha n^\beta} \tag{C.6}$$

This is a reasonable assumption, as the operational process of determining $w_V(n)$, would be the same as before when determining $w(n)$. So α and β are still determined by population sampling, and just as before, the more the mutations move the mutational clone away from the wild-type(which is the optimal solution), the more the fitness decreases. Before in (29), α and β were determined by minimising the difference between the population sampling of $w(n)$, and the estimation $|D|^{-\alpha n^\beta}$. This same process can be used to determine $w_V(n)$, as shown below:

$$\sum_{1}^{n} \left(|D|^{-\alpha n^\beta} - w_V(n) \right)^2 \tag{C.7}$$

This has been tested experimentally with the simulation. Below is the $w_V(n)$ samples at one generation.

A	B	C	D	E	F	G	H
|D|	lv	n	wv(n)	|D|^(-a*(n^b))	a	b	(wv(n)-|D|^(-a*(n^b)))^2
15	8	1	0.06667	0.06667	0.99999	0.86188	2.1669544984e-12
15	8	2	0.00417	0.00729	0.99999	0.86188	0.00000973781
15	8	3	0.00278	0.00093	0.99999	0.86188	0.00000341509
15	8	4	0.00208	0.00013	0.99999	0.86188	0.00000381246
15	8	5	0.00167	0.00002	0.99999	0.86188	0.00000271402
15	8	6	0.00139	0.00000	0.99999	0.86188	0.00000192074
15	8	7	0.00119	0.00000	0.99999	0.86188	0.00000141489
15	8	8	0.00104	0.00000	0.99999	0.86188	0.00000108558
						sum(H2..H9) =	0.00002410059

Figure C.1: Table of α & β Calculation by Minimisation of (C.7)

The $w_V(n)$ values were calculated from population sampling, and $|D|^{-\alpha n^\beta}$ (column E in Figure C.1 from left) was calculated by minimising the summation (C.7). A graph is shown on the next page of the mutational distance n against $w_V(n)$ and $|D|^{-\alpha n^\beta}$.

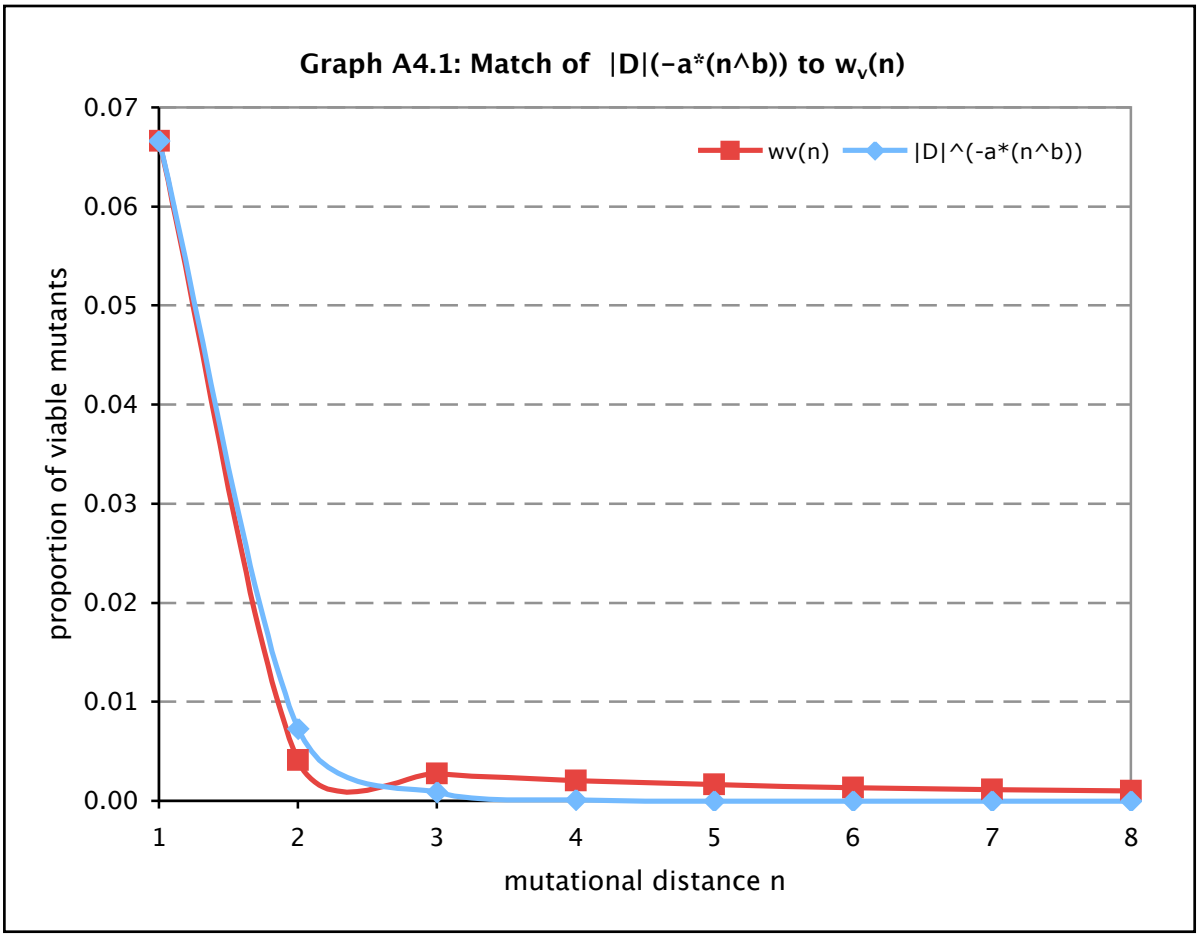


Figure C.2: Graph C.1: Match of $|D|^{-\alpha n^{\beta}}$ *to* $w_V(n)$

The match is strong, indicating that the $|D|^{-\alpha n^{\beta}}$ is a good estimate of $w_V(n)$. The exact form of $w_V(n)$ remains unknown, but α and β can be determined by minimising the difference in the population sampling and the estimation $|D|^{-\alpha n^{\beta}}$. So, C_{V_e} can be determined in terms of α and β:

$$
\begin{aligned}
&\textit{in } (C.1) \textit{ we found that } C_{V_e} = \ell_V - H(S|e),\\
&\textit{in } (C.2) \textit{ we found that } H(S|e) = \log_{|D|} |S_{viable}(\ell_V)|\,,\\
&\textit{in } (C.5) \textit{ we found that } w_V(\ell_V) = \frac{|S_{viable}(\ell_V)|}{D^{\ell_V}}\\
&\textit{in } (C.6) \textit{ we found that } w_V(n) = |D|^{-\alpha n^{\beta}}\,,\\
&\textit{and given } n = \ell\\
&\textit{hence}: \;\; w_V(\ell_V) = |D|^{-\alpha \ell_V^{\beta}}
\end{aligned}
\tag{C.8}
$$

$$
\begin{aligned}
\textit{and hence}: \quad C_{V_e} &= \ell_V - H(S|e) \\
C_{V_e} &= \ell_V - \log_{|D|}\left(w_V(\ell_V)\,|D|^{\ell_V}\right) \\
C_{V_e} &= \ell_V - \log_{|D|}\left(|D|^{-\alpha\ell_V^{\beta}}\,|D|^{\ell_V}\right) \\
C_{V_e} &= \ell_V - \left(-\alpha\ell_V^{\beta} + \ell_V\right) \\
C_{V_e} &= \alpha\ell^{\beta} \qquad \text{(C.9)}
\end{aligned}
$$

Although (C.9) is not finalised, it is a reasonable assumption, and was used to calculate C_{V_e} in Graph2 (Figure 26), which is duplicated in Figure C.3 below.

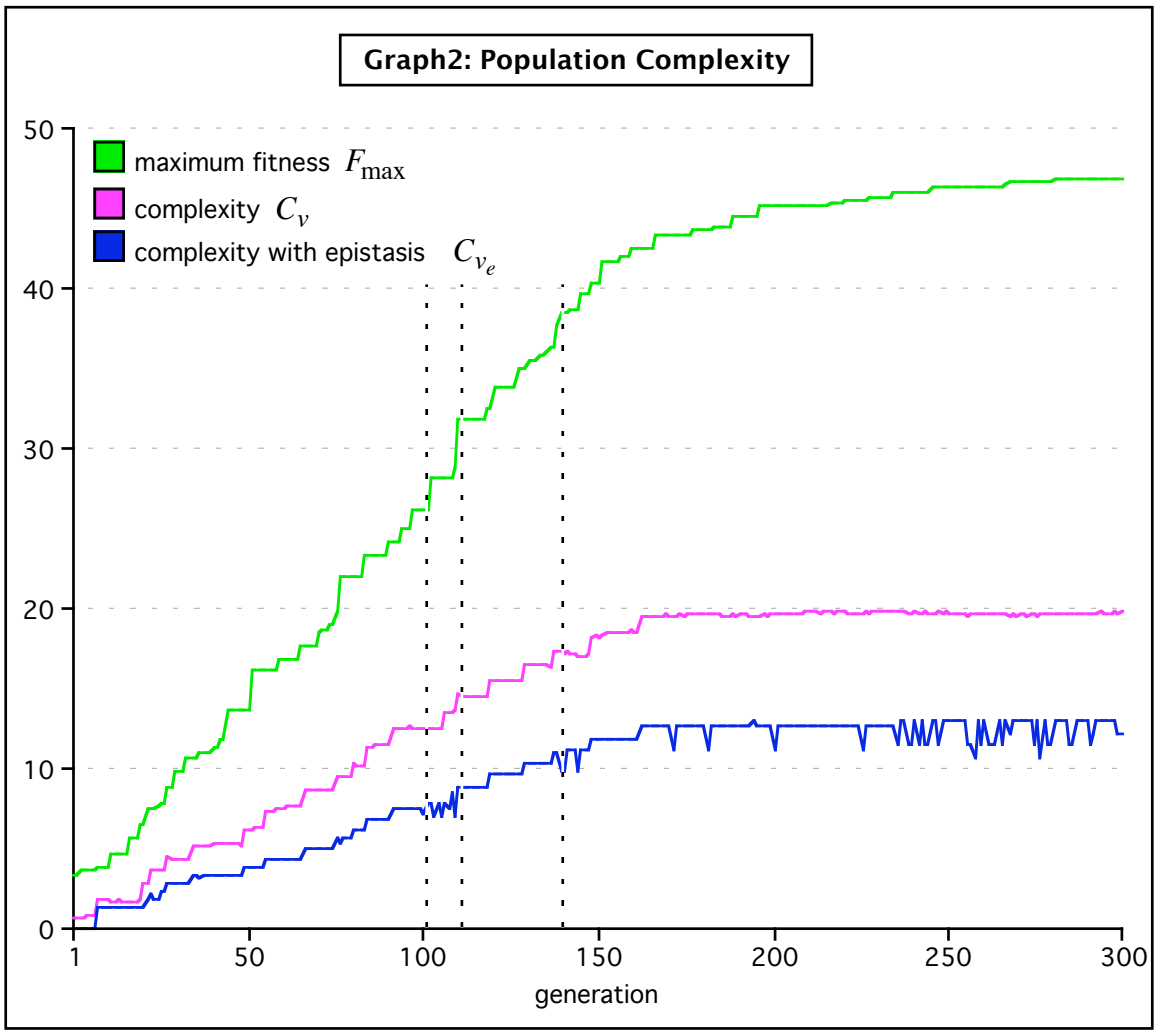


Figure C.3: copy of Graph 2 from Figure 26

The results appear to be accurate from the point of view that some epistasis is expected. The increasing epistasis over the generations is due to the increasing length ℓ_V of the population. With increasing length ℓ_V, there is an increased chance of epistatic effects because, it occurs between sites.

However, the level of epistasis once the optimal length $\ell_V = 20$ is reached, intuitively seems high. This could be a consequence of the formulation of (C.8) not being finalised,

or the difficulty in choosing wild-types which are not epistatically linked, for calculating α and β.

After the reformulation of (C.8) is finalised, if the epistasis remains as high, then we can be more certain that the choice of wild-types, which is already causing the erratic behaviour, is also estimating the epistasis too high.

It should be noted, that this method has previously only been used for corrective purposes[5], and not as an alternative complexity measure, precisely because of the difficulty in choosing wild-types which are not epistatically linked, especially in a population without recombination to minimise the epistatic effects.

D Presentation Slides

These slides were presented at the recent STU Mozart meeting, and were used to introduce the work in this deliverable.

agent = service/SM

- executable component (SDL interface)
- semantic descriptive component (BML)

MODELLING ASSUMPTION

DBE services = next gen web services

next gen web services merging with software agents

DBE services = software agents

5

Problem ?

De?ne self-organisation for a population of evolving agent-chains.

6

Population (of solutions)

agent

agent-chain

population (group) of agent-chains

7

Problem ?

De?ne self-organisation for a population of evolving agent-chains.

8

Population (vector of agent-chains)

Biology

Multi-Agent System

9

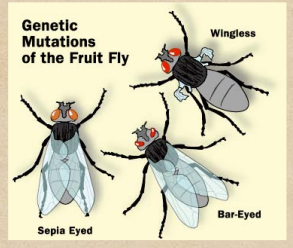
Replication (duplication)

Biology

Multi-Agent System

10

Mutation (point mutation)



Biology

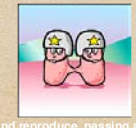
Multi-Agent System

11

Selection Pressure (User request in BML)

Antibiotics attack and kill off bacteria without the mutation...

The mutated ones survive after the antibiotics are gone...



...and reproduce, passing along the mutation to their offspring...

Eventually there are more antibiotic-resistant bacteria than non-resistant.

12

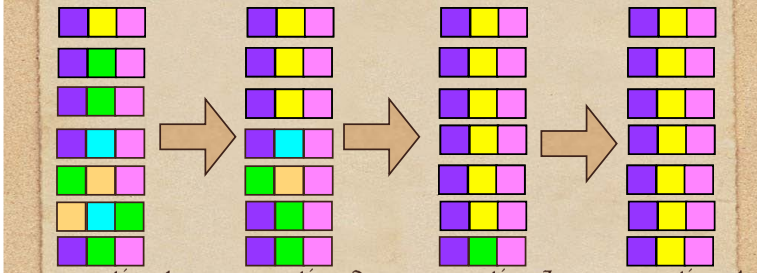
Problem ?

De?ne self-organisation for a population of evolving agent-chains.

generation 1

13

Organisation (of solutions)



generation 1 generation 2 generation 3 generation 4

14

Problem ?

De?ne self-organisation for a population of evolving agent-chains.

generation 1

15

Philosophy of Organisation

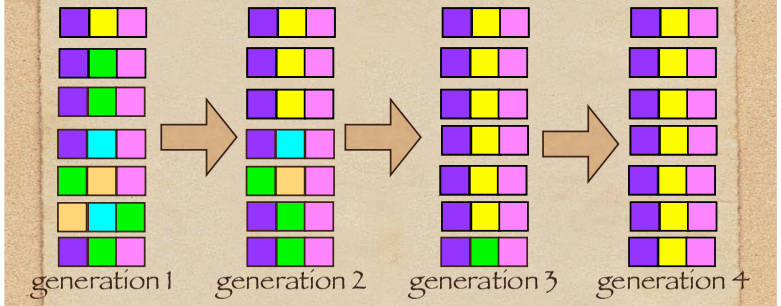
- Perspective - relative - to ?tness function
- Selection OR Selection Free

Philosophy of Self

Organising force is internal or external to the system.

16

Organisational Complexity



17

Entropy

site s = 1 2 3

generation 2

18

Entropy

$$H(s) = -\sum_{d \in D} p_d(s) \log_{|D|} p_d(s)$$

given : $s = site$

$\ell = length$

$D = alphabet$

$|D| = alphabet\ size$

where : $1 \le s \le \ell$

$0 \le p_d(s) \le 1$

$\sum_{d \in D} p_d(s) = 1$

19

Entropy

$$H(s) = -\sum_{d \in D} p_d(s) \log_{|D|} p_d(s)$$

s = 1 2 3

generation 2

given: D = alphabet = {■,□,■,□,■,□}

$|D|$ = alphabet size = 6

$1 \le s \le \ell$ ℓ = length = 3

$p_{■}(2)=0$ $p_{□}(2)=3/7$ $p_{■}(2)=0$

$p_{□}(2)=1/7$ $p_{□}(2)=1/7$ $p_{■}(2)=2/7$

$$H(2) = -\sum_{d \in D} p_d(2) \log_{|D|} p_d(2)$$

$$H(2) = -\left(\begin{array}{lll} p_{■}(2)\log_{|D|} p_{■}(2) & + & p_{□}(2)\log_{|D|} p_{□}(2) \ + \\ p_{□}(2)\log_{|D|} p_{□}(2) & + & p_{■}(2)\log_{|D|} p_{■}(2) \ + \\ p_{□}(2)\log_{|D|} p_{□}(2) & + & p_{■}(2)\log_{|D|} p_{■}(2) \end{array} \right)$$

$$H(2) = -\left(\frac{3}{7}\log_6 \frac{3}{7} + \frac{1}{7}\log_6 \frac{1}{7} + \frac{1}{7}\log_6 \frac{1}{7} + \frac{2}{7}\log_6 \frac{2}{7} \right)$$

$$H(2) = -\left(-\frac{1}{4} + -\frac{1}{4} + -\frac{1}{4} + -\frac{1}{4} \right) = 0.713$$

20

Entropy

$$H(s) = -\sum_{d \in D} p_d(s) \log_{|D|} p_d(s)$$

s = 1 2 3

generation 2

$H(3)=0$

$$0 \le H(s) \le 1$$

$$I(s) = H_{\max}(s) - H(s)$$

$$I(s) = 1 - H(s)$$

Information

21

Information

$$I(s) = 1 - H(s)$$

s = 1 2 3

generation 2

$H(2) = 0.713$

$$I(2) = 1 - H(2)$$

$$I(2) = 1 - 0.713$$

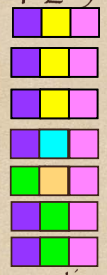
$$I(2) = 0.287$$

22

Population Complexity

sum of information

s = 1 2 3



generation 2

$C=2.06$

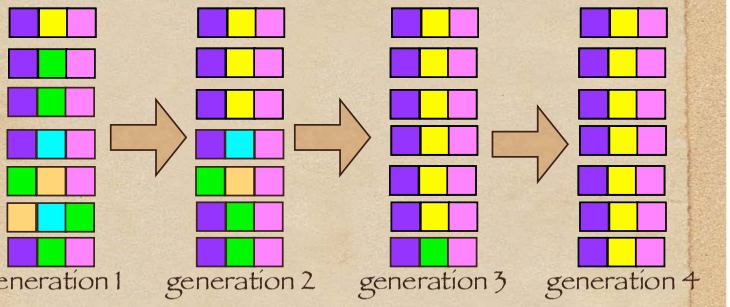
$$C = \sum_{s=1}^{\ell} I(s)$$

$$C = \sum_{s}^{\ell} (1 - H(s))$$

$$C = \ell - \sum_{s=1}^{\ell} H(s)$$

23

Population Complexity



generation 1 | generation 2 | generation 3 | generation 4

$C=1.62$	$C=2.06$	$C=2.77$	$C=3$
$E=0.54$	$E=0.69$	$E=0.92$	$E=1$

24

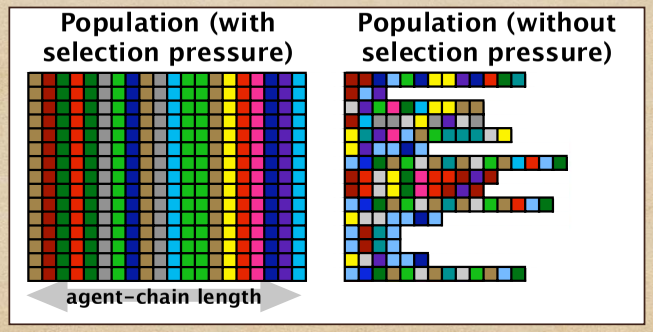
What next ?

- reformulate measure for variable length populations
- formulation of the ef?ciency E to indicate clustering within a population
- etc, etc, etc....
- make a SIMULATION with PRETTY PICTURES!

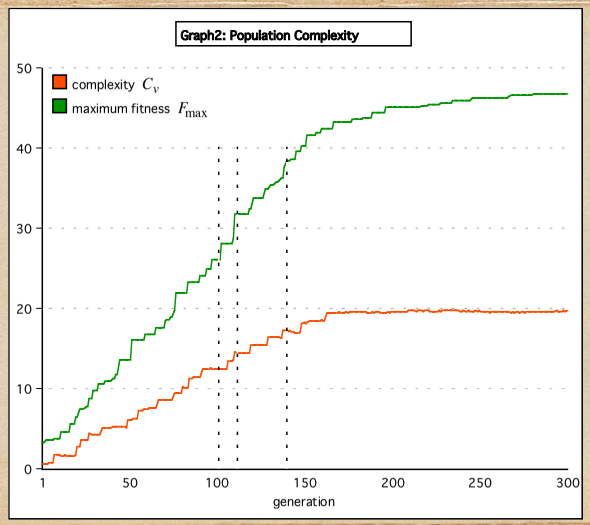
25

Visualisation

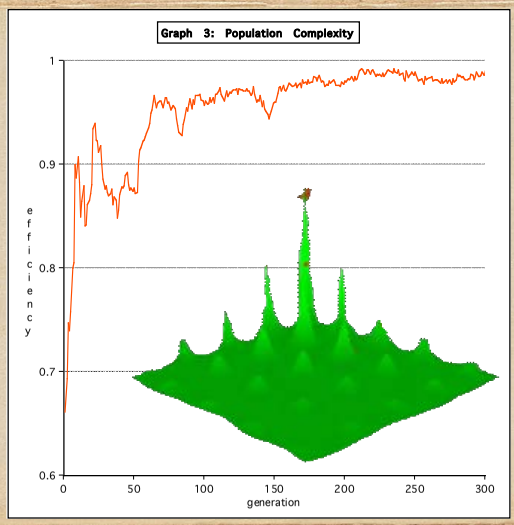
single agent/'DBE Service'
agent-chain/SM-chain



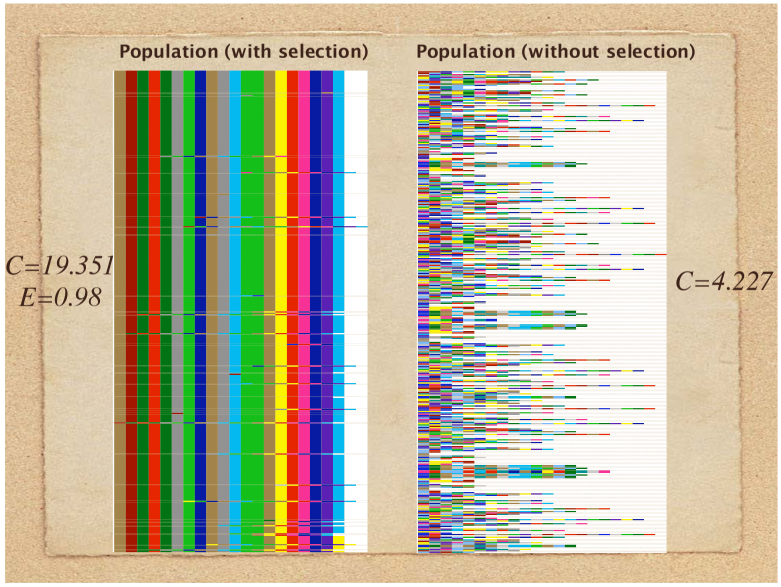
26



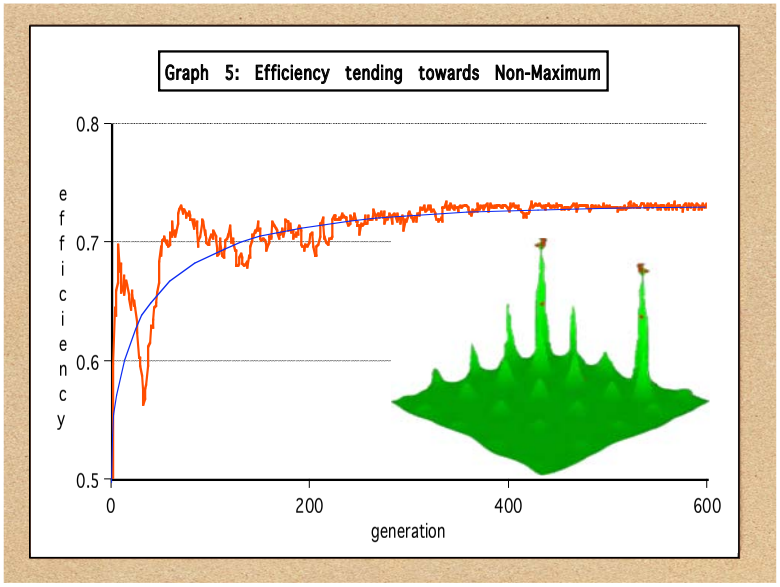
27



28



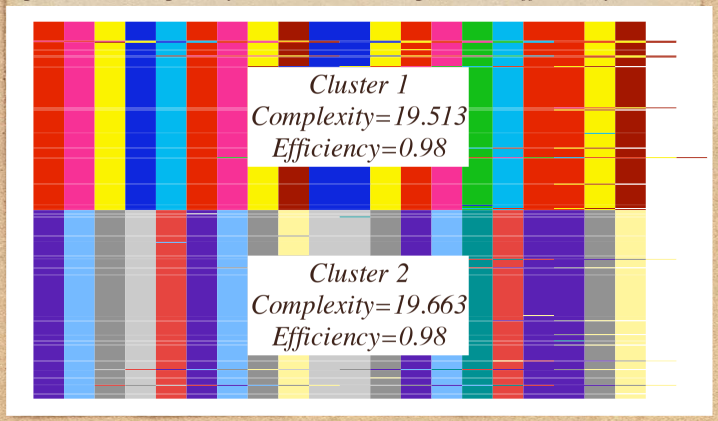
29



30

Visualisation:Clustering

PopulationComplexity =14.630 *PopulationEfficiency =0.7315*



31

Future work

- D6.1 Deliverable deadline
- extending the measure to the ecosystem
  - reformulation
  - comparison of ef?ciency
  - thermodynamical interpretation
    - information=matter, CPUtime=energy, money=?

Any Questions? Any Suggestions?

32