Illuminating meaningful diversity in complex feature spaces through adaptive grid-based genetic algorithms

Peter Charles Overbury

A thesis submitted for the degree of Doctor of Philosophy

University of Sussex

Department of Informatics

March 2020

Declaration

I hereby declare that this thesis has not been, and will not be, submitted in whole or in part to another University for any other academic award. Except where indicated by specific stated in the text, this thesis was composed by myself and the work contained therein in my own.

signature .....

Peter Charles Overbury

University of Sussex

PETER CHARLES OVERBURY

THESIS SUBMITTED FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Illuminating meaningful diversity in complex feature spaces through adaptive grid-based genetic algorithms

SUMMARY

In many fields there exist problems for which multiple solutions of suitably high performance may be found across distinct regions of the search space. Optimisation of the search towards including these distinct solutions is important not only to understanding these spaces but also to avoiding local optima. This is the goal of a type of genetic algorithms called illumination algorithms. In Chapter 2, we demonstrate the use of an illumination algorithm in the exploration of networks sharing only a given set of structural features (valid networks). This method produces a population of valid networks that are more diverse than those produced using state of the art methods, however, it was found to be too inefficient to be usable in real-world problems. Additionally, setting an appropriate resolution of the search requires some amount of prior knowledge of the space of solutions. Addressing this problem is the focus of Chapter 3, in which we develop three extensions to the method: a) an exact method of mutation whereby only valid networks are explored, b) an adaptive mechanism for setting the resolution of the search, c) a principle for tuning mutations parameters to the search' s resolution. We show that with these additions our method is able to increase the diversity of solutions found in significantly fewer iterations. Finally, in Chapter 4 we expand our method for use in more general problem spaces. We benchmark it against the state of the art. In all tested landscapes, we show that our method is able to identify more meaningful niches in the spaces in the same number of iterations. We conclude by highlighting the limits of our framework and discuss further directions.

Acknowledgements

I would like to briefly thank all the people that made this work possible. From its practical development to the development of me personally as a researcher, this would not have been possible without these people and I am truly thankful. I have always been aware of the tremendous amount of effort expended in not only keeping me up-right but wondering.My hope is that with this work I present here I have proven all of these peoples efforts worth the trouble. As such here are just a few of these people by name:

My supervision team Prof Luc Berthouze and Prof Istvan Kiss, whose guidance have taught me the highest quality of research standards.

Dr Mario Pannunzi for his support and guidance, and Harry Collin for his work on the tree storage used in this project. Thanks for putting up with all the spare parts and random rants that got me through this process.

My family, particularly my dad, and all my friends, especially Ed, who helped a lot with the keeping me upright part.

Sussex and the hardship fund for their help with proofreading expenses and Sussex in generally for the many place to sit.

The IISP lab, not only for my new post doc role but for the patients in letting me finish this work.

Sophie Maskell, you know what you do. Please keep doing it for ever..

Contents

Chapter 1

Introduction

In almost every field of science and engineering there exist problems for which the space of possible solutions (often called the search spaceor problem landscape) is far too large for a systematic search of the space to be realistic within current computational limits [47, 155]. Furthermore, many of these problems require realisation via a physical system or via complex simulations to determine their performance or behaviour. This makes the relationships between the variables being changed and their outputs even harder to fully comprehend, and leads to the accruing of additional computational cost for the search of the space.

One strategy for investigating these spaces is to try to sample the space of solutions in as unbiased a fashion as possible, either via random sampling of the space [27] or via more advanced methods, such as Markov chain Monte Carlo simulations [13]. The issues with such approaches are that: (1) they often require very long mixing times to gain a good comprehension of the space; and (2) we often care more about exploring only a particular part of the space, rather than the entire space of solutions. With respect to the latter, some regions of the space might not yield realisable solutions when they are simulated to detriment their performance. Alternatively, there may be poor quality solutions (i.e., low performance) or some regions might not be different enough from one another to be worth investigating separately,

as their effects on the variable of interest are very similar and/or identical. Thus for the majority of these problems, "black box" optimisation algorithms are often used due to their ability to automatically find one or a set of high-quality solutions within these large spaces. This focusing of the exploration, or "search", of the space of possible solutions allows particular questions between the variables being changed and the maximum performance (also called the "fitness" of the solution)possible in the space to be investigated. This occurs without the need for full coverage of the space (which results in significantly lower computing costs) or explicit information about the relationship between genome variability and performance. As such, these algorithms are often able to out-compete handcrafted solutions, particularly in problem spaces in which the relationship between the features and performance is not well understood, such as in the material sciences [57, 107, 65], in drug discovery [183, 103], and in many other domains [85]. However, this focusing of the search towards optimal solutions can yield a restricted view of the space of possible solutions. This can even lead to global optima (the highest fitness solutions possible in the space) being missed altogether as the algorithms are deceived into converging onto local optima, because they are unable to see enough of the space to know about the global optima [97, 136]. These kinds of "deceptive landscapes" (i.e., those in which there are one or more dead-ends in the search space with respect to increasing the value of the objective function [92]) highlight the need for a re-balancing of these algorithms towards more exploration (i.e., the focusing of the search towards areas of the space not yet sampled, sometimes referred to as the "novelty" of the solutions). This may even occur at the cost of exploitation (i.e., the focusing of the search towards the areas of highest fitness seen), so as to gain the best understanding of the space. However, even when the search is only exploring high-fitness areas, a level of exploration is still a key to gaining insight into these kinds of complex solution spaces and their relationship to particular measures of performance.

So how do we increase the exploration of these kinds of spaces without the large computational cost associated with a fuller search of the space. In other words, how do we insure good

coverage of the space of solutions while wasting the minimal amount of time on the areas of least interest to our investigation?

Inspiration for a solution to this question can be gained from an examination of the wealth of diversity produced by natural systems in solving many of the same kinds of problems [105, 86, 185, 56], and from bio-inspired techniques developed from these observations seen in these natural systems [47, 75]. Indeed, the study of the way that these systems "evolve" such solutions has given rise to the field of evolutionary algorithms (EAs), and more specifically genetic algorithms (GAs). These algorithms implement the methods seen in these kinds of natural systems in order to find the same kind of high calibre solutions in large complex unknown problem landscapes.

In the following sections, I will first summarise GA-based approaches used thus far to address this problem, as well as how effective they are (Section 1.1). Then, I will describe our proposed case study to explore the space of higher-order structure in graphs sharing fixed levels of global structural measures and justify why it is a good case study for this problem (Section 1.2).

1.1 Genetic algorithms in the exploration of unknown search spaces

GAs are a method of searching the space of solutions (i.e., a collection of possible solutions with some notion of "distance" between candidate solutions [21], sometimes also called the "genome space") that are inspired by the biological phenomenon of natural selection [33].

They were first proposed by Holland in the 1960s [63] as a way of studying the mechanisms of adaptation seen in nature and thus applying them to a wide range of problems [62, 87]. This includes, but is not limited to: automatic programming [147, 37, 36]; machine learning

[104, 85]; and of course a wide range of optimisation problems [46]. There is no rigorous definition of what makes up a GA [114]. Almost all of their properties can and have been altered in various ways in order to create variants that are best suited to the questions asked by their creators.

In their simplest form, GAs work by taking a starting population of individual solutions and encoding them so that each is represented by a "genome" that holds the key variables being studied/optimised. The choice of the encoding method for the GA plays one of the most important roles in defining the search space available to it [87]. For most optimisation problems, this will define the variables that will be optimised. However, when using a method of encoding that requires some form of realisation of the genome to yield a phenotype, this choice might significantly affect the kind of solutions available to the GA. In the majority of GAs, the population is initialised using a random sample of the search space, although this can be manipulated in order to gain better coverage of the starting space or higher quality starting solutions (often used as a means to avoid local optima [114]).The size of this population, if kept constant throughout the search, can affect the "memory" of the GA. This means that a smaller population of individuals is more likely to perform poorly within a deceptive landscape, because their representation of the space is based solely on the size of the population it is compared to [100]. This population is then "evolved" through the iterative process of selecting one or more individuals and changing their genome in some way; these are referred to as "genetic operations" [114]. These genetic operations are the method by which the GA explores the space of solutions, normally via: a "mutation" (adding to, or subtracting from, values in the genome [168, 181]); a recombination or crossover (combining two individuals into one new individual with a new genome [169]); or some combination of both [64, 134, 64]. The rate at which these genetic operations are applied and their size are often used as a way of altering the balance of exploitation vs exploration within the search (i.e., the larger/more frequent the genetic operations, the more exploration is achieved [64]). The genetic opera-

tions can also be used as a way of avoiding/encouraging movement in particular areas of the search space [148, 166]. They can even be dynamically changed during the search to alter these effects in response to external clues (such as a set number of iterations [126]) or internal features of the search (i.e., features gained during the search, such as passing a set level of performance or a set level of coverage of the search space [124, 98]). This is not an exhaustive list of possible genetic operations, as any change of the genome could qualify, provided that it involves some heritable feature of the genomes selected for genetic operations, in order to preserve any operational features of the genome (also known as "selective pressure" [114]) and to avoid comparisons with random search methods [64, 156]. All individuals, new and old, are evaluated for their fitness (the measure or measures by which all individuals in the population are compared). This sometimes involves the realisation of the genome via some physical or mathematical simulation to give the resulting behaviours or "phenotype" of interest. Importantly, the relationship between genome and fitness (often referred to as the "fitness landscape" [184]) can be complex. These evaluations of the fitness of the population are then used to affect how prolific the genome of each individual will be going forward, with the lowest fitness individuals either removed or selected for genetic operations less often. This results in a population that, depending on the setup of the GA, will move through the search space towards areas of highest fitness, creating the selective pressure [62]. This is perhaps the most flexible property across different GAs, with thousands of different methods for the following: selection (i.e., determining which of the individuals in the population is selected to be a parent to new solutions via genetic operations [156, 145]); population storage (e.g., storing only the top-performing percent of individuals or removing the bottom percent, which is termed "elitism" [185, 114], or using another method such as storing all solutions regardless of fitness); and/or other features used to control the selective pressure of the GA throughout the search. These features do not even have to involve the optimisation of fitness. Instead, they might be focused on creating all kinds of pressure within the search, pushing the population

to include, or more heavily focus on, particular aspects of the space, such as the diversity of the genomes in the population.

This cycle of applying genetic operations, evaluating the population and then updating the stored population (referred to as one generation) is repeated until the end goal is reached. This goal is usually a given number of generations or a given fitness level. The population of solutions combines with the iterative process of searching the space of solutions via competition or other selective pressures to inform the search of the space as more solutions are found during the evolution. This approach has made GAs well known for the novelty of their results. In particular, they are often able to find solutions that were never seen in the population before nor considered by the handcrafted solutions created by researchers [94, 178, 65, 125].

This does not imply that all GAs will overcome the problem of deceptive landscapes or that they will gain perfect coverage of all areas of interest in the space. Rather, they still require consideration of the problem landscape in order to ensure a greater coverage of the space and thus avoid areas of local optima.

Recently, work from a new school of thought geared towards abandoning goal-oriented fitness measures has started to show the effectiveness of directly measuring novelty (i.e., the extent to which solutions have not seen before in the population or sometimes not been seen in any past search of the space) as a fitness measure. This allows optimisations to be driven towards a goal without explicitly setting the GA that goal [92]. This differs from the methods discussed previously that encourage or preserve the diversity of solutions within the population of the GA; rather, the search is actively pushed towards areas of the space not seen before in the population.

One method that combines such a "novelty search" with other fitness measures is the Novelty Search + Local Competition (NS+LC) method [117]. In this approach, the novelty of a solution is determined by comparing it with its K nearest solutions in the genome space, although this can be changed to explore the diversity of other features as desired. This means

that each solution is only kept in the population if it has the highest fitness (as determined by the optimisation measure) compared with the K nearest solutions.Such approaches have been shown to be highly effective and have led to the development of the field of quality diversity [30, 7, 142, 141].

Both the performance measures being optimised and the novelty of a solution are treated as their own separate fitness measures. These are optimised simultaneously, in a process referred to as "multi-objective optimisation" [84, 165, 54]). By treating each measure as its own separate fitness, not only is diversity of the solutions increased, but, because the GA is guided by two linked feature spaces, it is less likely to be caught in local optima. A weakness of this method for exploration in a full search of the space is that it is still much more focused on the use of the novelty fitness to improve performance measures than on the exploration of the feature space as a whole. This makes it vulnerable to issues such as "novelty cycling", whereby the population merely moves from one area of diversity to another and back again without memory of where it has already explored [118].

As such, if the aim is not merely to find as diverse a population of high-fitness solutions as possible (as is the case with the kind of optimisation problems NS+LC was created for), but rather to identify all areas of high fitness across the space (and just provide a more accurate representation of the feature space), there is a need to focus on preserving diversity at a global level rather than just maximising it. Indeed, two points very far away from each other in feature space might yield high diversity in the population without a real representation of the feature space. In other words, just because two individuals are on the map (a local view) does not mean that anything can be inferred about the space in between them (which requires more of a global view of the space).

Tackling this problem led to the development of "illuminative algorithms", whose focus is on identifying the highest-performing solutions at each point of the space [118].

The first of these methods was the Multi-dimensional Archive of Phenotypic Elites (MAP-

Elites) method [118] (see Algorithm 1 for pseudo-code), in which the optimisation is treated as the single objective of the fitness measure, and the novelty of solutions within the population is promoted via the mechanism of elitism based on a grid-based storage of the population. This grid is created by dividing the search space (or, as for NS+LC, any feature for which diversity is desired) into "cells" of a given size. The grid is often set so that there is a given number of cells that are then used as "niches" for the population. It is worth noting that in the majority of cases the mapped space used for these cells is placed in some form of "behavioural space" witch while related to the genome space dose not have a one-to-one correlation in movement around the space. These mapping of the behavioural space are what define the difference from quality diversity (QD) algorithms and the broader illuminative algorithms talked about here. This "niching" of the population means that, during the search, solutions are compared only with other solutions that fall within the same cell. In other words, the first time a solution is found in a new part of the space (i.e., an empty cell), it will be stored regardless of fitness. If instead there have been other solutions in that cell (i.e., there has been "revisiting" of the cell) then optimisation will act within that cell and keep only the best solution.

This provides not only a fast way of assessing the novelty (and diversity) of a solution compared with the whole population, but also guarantees that if a new region of the space is discovered during the search, it will be preserved. Thus, unlike for NS+LS, the longer the search is run, the higher our confidence is in the coverage of the space by the population, and thus the improved representation of the space.

This is of particular importance for cases in which we are not sure of the location or distribution of the solutions in the unknown space. These approaches have been used effectively to generate a diverse range of solutions in a number of problem domains in which coverage of the space of the solutions is of great importance [30, 73, 79, 5, 58, 48, 186].

A more detailed breakdown of the MAP-Elites method is also provided in Sections 3.3 and 4.2. In these sections, we cover in more detail some of the limitations of this method, which

come from very rigidly defining the regions of diversity (i.e., what makes two individuals either side of a dividing cells boundary "diverse" from each other) and discuss the need for "meaningful" diversity. That is, going beyond simply measuring the given distance between individuals towards a meaningful difference in the way individuals function that is relevant to the questions being asked of the space. Before starting this PhD, I worked on the use of a NS style GA for the mapping of network structures sharing only the same global clustering and degree distribution, summarised in the paper [129].

The first publication included in this thesis (Chapter 2) tests the effectiveness of a MAP-Elites style method in investigating a large unknown space of solutions, namely the space of all possible network configurations sharing only a given degree distribution and global clustering coefficient.

In the following section, we briefly outline the importance of this problem space in the field of graph theory, the challenges faced when investigating these relationships, and the current methods available to tackle this problem.

1.2 Graph theory (networks)

Networks of graphs are a way of modelling almost any complex system that involves the interaction of individual components by representing the connections (the "edges" or lines) between each individual part of the system (the "nodes" or vertices; see Figure 1.1). This modelling often allows global behaviours dependent on the connections and/or relationships between different elements of the system to be picked out that either would not have been noticed in isolation or that would be undetectable within large data sets [119].

These network models can be further tailored to represent features of systems. For example, a network can be directed, in which case all edges are given a direction denoting the flow of information (see network *a* in Figure 1.1). Weighted networks can also be used, in which

weights are given to each of the edges that represent the amount the connection contributes to the flow of information (also see network *a* in Figure 1.1). These propensities have resulted in applied network theory being pervasive in real world applications, including the spread of disease [59], of information via social media [137]), and of neural information within the brain [109].

The structure of networks is often represented by a N×N matrix (called an adjacency matrix), where N is the number of nodes in the network and each column and row shows the connection between each of the two nodes (see Table 1.1). Here, a 0 denotes no connection between nodes, while a 1 means a complete connection. When dealing with networks with no self-connections (i.e., a connection from a node to itself; see node 3 in network *b* in Figure 1.1), there are always zeros across the diagonal of the matrix. If all of the connections are non-directed, the matrix will also be symmetrical (see the matrix of network *c* in Table 1.1).

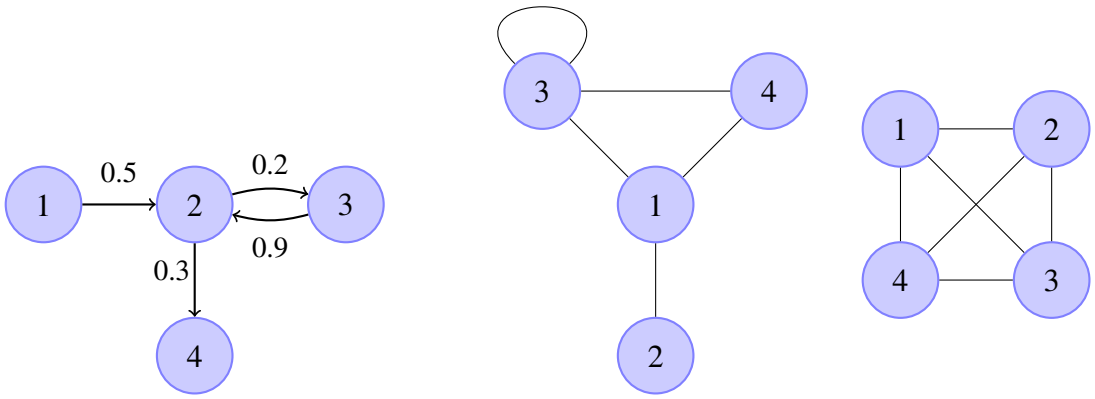


Figure 1.1: Examples of networks containing four nodes (represented by circles), with the connections either undirected (lines without arrows) or directed (lines with arrows). (a) A directed and weighted network. (b) An undirected network. (c) A complete and undirected network.

From these adjacency matrices, we are able to extract and study various structural properties of the networks. Often these studies focus on properties that convey useful information about how the global structure of the network affects the operations of dynamical processes in the system represented by this network [10]. These properties include, but are not limited to, global measures of network structure. One such property is the degree distribution, which is the frequency distribution of the number of edges of each node (their "degree") in the network [119]). Another is the global clustering coefficient, which is the ratio of closed triplets (three nodes connected by three edges) to the number of closed and open triplets (three nodes connected by two edges, i.e, $\frac{closed}{closed+open}$ [122, 179]). A further property is assortativity, which is the preference of a node to be connected to other nodes that are similar to itself in some way (for example, those nodes with a similar degree [119]). However, there are also other properties that can involve more local structures (also sometimes referred to as higher-order structures). These include "motifs", which are small network structures that are found to occur statistically significantly more often than expected at random [4, 113], such as ◹ (also known as "Toast"; see Figure 2.1), and "subgraphs", which are similar to motifs, being small network structures within larger networks, but that are arbitrarily structured and sized [71] (see all panels in Figure 2.1).

Assuming that these structures, which are seen in real world networks, act in some way to modulate the efficiency of the dynamics or functions of these networks (a common belief in the field [121, 19, 115]), then it would be valuable to study the extent to which it is possible to optimise networks by manipulating these structures. This makes the interdependency (i.e., how dependent feature X is on feature Y) between network properties and their structures of great importance to the field [164, 51, 26]. This is because being able to explorer a studied property's independence from the other properties of the network is vital to determining which properties truly affect the dynamics seen in the real world network.

Not surprisingly, this is a well-researched problem [24, 6] and the current standard for

	Nodes (V)	Edges (E)	Adjacency matrix	
			in	out
a	{1,2,3,4}	in={(1,2),(2,4),(2,3),(3,2)} , out={(2,1),(2,3),(3,2),(4,2)}	$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0.9 & 0 \\ 0 & 0.2 & 0 & 0 \\ 0 & 0.3 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0.2 & 0.3 \\ 0 & 0.9 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$
b	{1,2,3,4}	{(1,2),(1,3),(1,4),(3,4),(3,3)}	$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$	
c	{1,2,3,4}	{(1,2),(1,3),(1,4),(2,3),(2,4),(3,4)}	$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$	

Table 1.1: The network structures shown in Figure 1.1: (a) A directed and weighted network. (b) An undirected network. (c) A complete and undirected network.

addressing this problem is to generate many networks sharing only the studied property; that is, they should in all other respects be random in their structures (these are referred to as "null models"). The generation of null models is in itself a non-trivial problem, as the solution space of possible network configurations for networks of any realistic/useful size (1,000 nodes or greater), is extremely large. This is the case even when the solutions are restricted to those sharing the desired structural property or properties. This implies, as discussed in Section 1, that it is unfeasible to fully map the space of solutions via exhaustive sampling.

Not surprisingly, the majority of methods designed to generate networks sharing global measures, such as a fixed level of global clustering and degree distribution, neglect to control for changes in more local measures and thus might not produce true null models [82, 149, 80, 130]. This is of concern, as research has shown that these higher-order structural properties will fundamentally affect the way the system operates [149, 150, 130, 131]. As such, there exists a need for accurate methods of generating networks given global features that provide more control over the exploration of higher-order structures.

This makes this problem a prime candidate for testing the kind of MAP-Elites-style GAs described in Section 1.1, which provide greater diversity and novelty in the population of networks generated. In fact, GAs are not uncommonly used in the optimisation of network structures [90, 15].

Although they are normally used for optimising the processes run on the network, rather than for exploring the structures of the network [91, 17, 160], GAs have been shown to be effective in the space of networks sharing fixed global clustering and degree distribution [129, 130] (to our knowledge, this paper still contains the only method put forward for using GAs to explore the structural diversity of networks sharing only fixed structural features, not counting those methods presented in this thesis).

Degree distribution is perhaps the most commonly constrained structural measure for generating these kinds of null models [113, 127], as it is one of the most fundamental properties

of a network [119]. It has been shown to be an important feature of many real world networks, such as the scale-free network of the brain [43].

Combined with the global clustering of a network, which is a major feature of many real world networks [159, 108], these two properties are often considered enough for the description of the structure of a network [150, 120]. However, there is growing evidence that this is not the case when considering higher-order structures [150, 149] and furthermore, that the methods used in the generation of null models of a given global clustering and degree distribution do not fully sample the space of higher-order structures existing in the search space [129, 130].

In the following section, we describe the limitations of current approaches for generating networks of fixed degree distribution and global clustering. We additionally highlight previous work using GAs to increase the diversity of the networks produced.

1.2.1 Network generation methods

The problem of generating null models for a fixed level of global clustering and degree distribution (referred to as a "valid" network henceforth) is non-trivial and involves many layered problems.

For example, in order to constrain the degree sequence, the presence of self-loops (connections for which an edge starts and ends with the same node, "looping" back onto itself) and multi-edges (where more than one edge connects node A to node B) becomes a problematic issue to overcome [119].

Currently, there are two broad types of methods used for generating networks with a set level of features: "rewiring" and "direct construction" [34, 119].

Rewiring (also called "edge swapping") methods involve taking a starting network, normally of the same size and number of edges [11], and then swapping pairs of edges repeatedly

using a Markov chain scheme (i.e., a sampling the space of solutions via a walk in which the probability of each step is dependent only on the state attained in the previous event [152]) to produce new graphs with the same degree sequence. Such methods can result in self-loops and multi-edges, but these can be removed by only performing the "swap" if it will not induce self-loops or multi-edges.

This process repeats for $E \times Q$ time steps, where E is the number of edges in the graph and Q is chosen to be large enough for the Markov chain to show good mixing (with non-swaps due to self-loops or multi-edges being counted in the overall number of repeats) [143]. Thus, when looking for a set degree sequence or degree distribution one merely adds the constraint of a set degree to the required checks before a switch can be confirmed [110]. In other words, the change must preserve the degree and should additionally not lead to self-loops or multi-edges.

To further constrain the networks produced by this kind of method and to gain a given level of global clustering (i.e., the number of closed triangles as compared to paths of length 2), we can guide the choice of edges to be "swapped" to increase clustering. One such method is called "Big-V", in which rewiring is done via the selection of a chain of five distinct nodes (see Figure 1.2) that are then broken into one triangle and one disconnected pair; the rewiring is then only accepted if it increases the clustering of the network [67, 59, 68].

A limitation of this kind of method is that (due to the requirement of a starting network to either preserve the network features controlled for or increase the property from a starting network), such methods normally have high dependency on the starting network selected. This bias can be lessened by using multiple starting networks and combining the pool of rewired networks [3] created, but this does not guarantee the diversity of the pool of networks produced. Furthermore, gaining a complete sampling of the space, (i.e., samples from a diverse range across the search space) would require very long mixing times. Moreover, this still does not address the diversity of higher order structures within these generated networks,

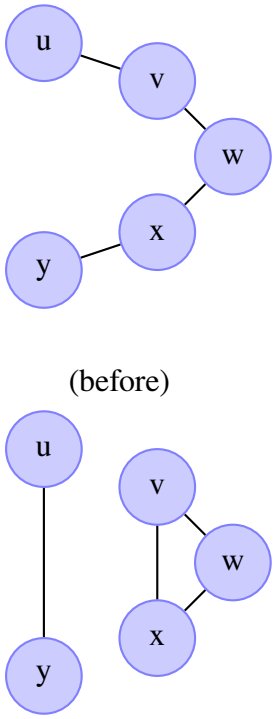


Figure 1.2: Illustration of the operation of BigV. Top: five distinct nodes in a chain (before) are selected by BigV to have their edges swapped. Bottom: an example of a swap that would be accepted (after), i.e., one that increases clustering.

which does not differ greatly from that seen in the starting networks [151].

We sought to improve this random walk method of exploring the space of possible networks, in which there is no direction to the movement through the search space, other than ensuring that each "step" produces a valid network.To address this, a GA using a rewiring-like method of mutation was employed and was found to be effective in improving the diversity of samples taken from the space of solutions. This occurred without the need for an exhaustive search of the space compared with popular re-wiring methods such as BigV [130].

The work of [129] was in fact the basis of my master's thesis and served as a preliminary investigation into the feasibility of this kind of novelty-driven GA for exploring higher-order network structures. In this work, networks were encoded via their adjacency matrix and mutations were conducted at an edge level, similar to the rewiring methods mentioned earlier in this section. These matrices were then optimised toward two fitness measures: (1) structural fitness (the distance between the target and actual degree distribution and global clustering, with the fitness based on minimising the distance to the target); and (2) novelty fitness (the difference in edges between the selected networks and all other networks in the population with a structural fitness of 0, i.e. valid networks). Storing this pool of valid solutions found during the evolution gave the GA a kind of memory of where it had been, thereby preventing novelty cycling. As a result, this work was able to find significant diversity in a range of different network types, including K-regular, Poisson and normal degree distribution networks with a range of clustering values, and even in very small networks in which the level of diversity was often inherently limited by the small number of connections. However, this method was incredibly wasteful in the way it found networks, both because of very imprecise mutations that resulted in invalid networks most of the time, and because of a lack of scalability. Indeed, it was only feasible to generate very small networks (< 200 nodes) and even then, obtaining a pool of around 50 valid networks took 2 days of running time on the high-performance cluster at the University of Sussex. This therefore prevented this method being considered a realistic

alternative to current network generation methods. However, it did show the feasibility of using a GA to explore the diversity of network structures.

The second broad approach is direct generation, also called "stub matching". These methods are unlike rewiring, in that they do not require a seed network. Instead they build graphs by sequentially matching two "stubs" that represent a potential connection or "dangling half-edge" that can be paired to make one edge [119]. One of the best known, and most referenced, examples of such method is the configuration model [123], in which an arbitrary degree distribution is achieved via the random matching of all of the stubs that are not yet connected. Any occurrence of self-loops or multi-edges is then dealt with by restarting the whole process again from the start.

Alternatively, this method can be improved through methods such as start-constrain [112], in which one node is selected as a "hub" and an "allowed" and a "forbidden" list are made for the hub, before nodes are randomly picked and connected to it until the network is completed. The allowed list included all nodes that can connect while still preserving graphicality (i.e., here meaning the network is realisable without self-loops or disconnected nodes), while the forbidden list includes all nodes with non excess degree (i.e., those that cannot be linked with the hub anymore, starting with just the hub itself and increasing with each link). This does not imply that this method is capable of replicating all possible networks with that constraint, nor that this will extend to further constraints on the network to be generated. In order to further constrain the network it is possible to modify the standard configuration model [123], such that it constructs networks using specific network subgraphs.

One such method is the Clustered Configuration Model (CCM) [77], in which a set level of clustering is achieved via the following looping method:

1. Allocate a number of stubs following a given degree distribution to a node;

2. Multinomially determine the configuration of corners and single stubs;

3. Create lists for each corner type where a node that is allocated K corners of a certain type will appear K times in the corresponding corner list;

4. Draw corners at random and without replacement from the appropriate lists and connect them with other corners to form motifs;

5. Repeat until all lists are empty.

Again this method must deal with self-loops and multi-edges by restarting the whole process. However, as this will depend only on the degree, this will become negligibly small in the limit of a large network, such as those most often studied in the field [151] (i.e., networks of approximately 10,000 nodes). This of course can mean that the generation of some graphs might have an very long computational running times, depending on the occurrences of self-loops or multi-edges. Variants of the method have been suggested that involve leaving such occurrences unmodified until the end to be fixed by hand. However, this has been shown to cause biases, even in the limit of large network sizes [82].

Recently, the CCM has been extended to allow the generation of networks with any arbitrary subgraphs. The cardinality matching algorithm (CMA) [151] is discussed in detail in Sections 2.3.1 and 3.3.1. This method allows for much tighter control of the networks produced and means that particular features of the network structure can be more easily explored, while still maintaining the degree distribution and global clustering set. However, on their own, these methods do not have any mechanism by which to systematically sample the solution space.

Therefore, we propose to improve upon our previous work [130] by using the direct generation method discussed above to encode the networks. These can then be mutated at a subgraph level by increasing/decreasing the counts of the different subgraphs in the genome, and then evaluated for their level of global clustering via their realisation using the CMA direct generation method [151]. This allows the exploration of diversity with more direct control

over higher-order structures (i.e., the subgraphs chosen for the encoding), as well as enabling more efficient and scale-free storage of networks (i.e., mutation and storage of these networks is not affected by their size or number of connections).

This method and its results is the focus of Chapter 2 and is covered in full there.

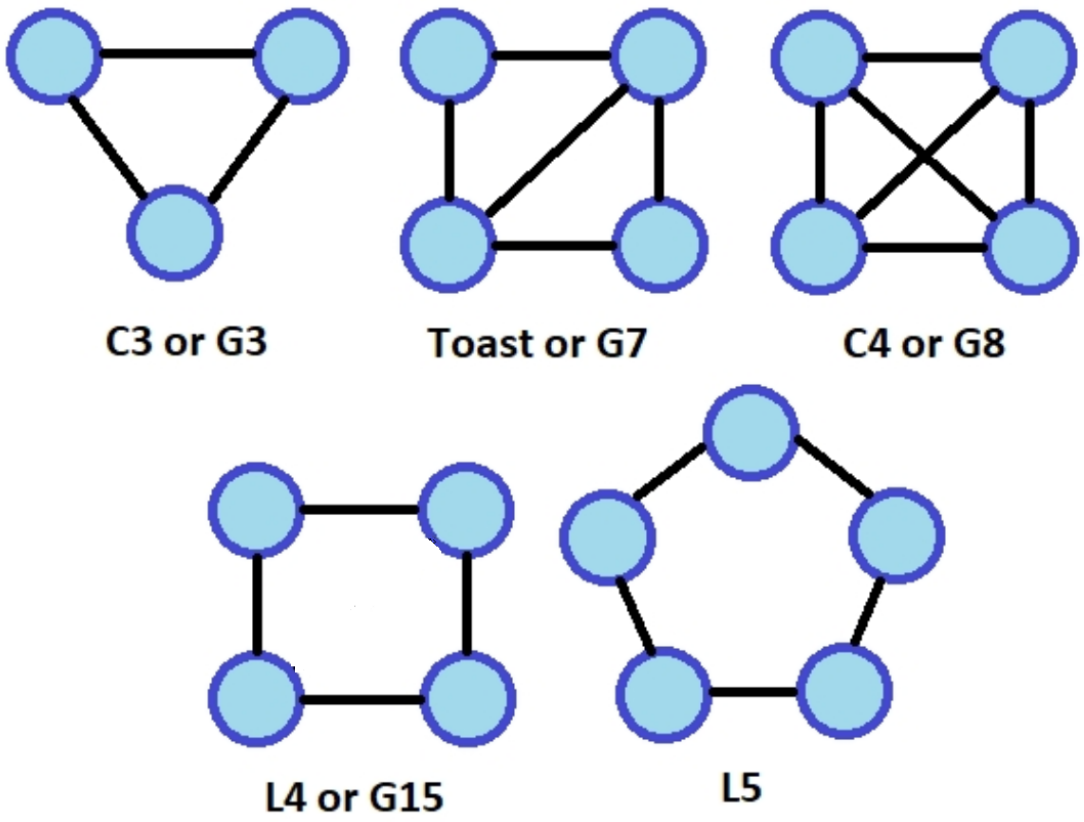


Figure 1.3: Examples of subgraphs used in this thesis. All are assumed to be undirected and unweighted. From left to right, starting with the top row, they are: triangle (C3 or G3); partially connected square (Toast or G7); fully connected square (C4 or G8); empty square (L4 or G15); and empty pentagon (L5).

1.3 Structure of the thesis

This thesis seeks to improve the state of the art in the mapping of diversity and novelty of solutions produced from the exploration of massive unknown search spaces. This work is applied to the mapping of complex landscapes not seen before in this field, and to progressing the field further towards the use of "meaningful" diversity between samples (i.e., such that the level of change in any one examined feature results in a meaningful difference in the investigated behaviour or feature).

To begin with, the mapping of these search spaces is often far too computationally taxing to use exhaustive search or even other random sampling methods (e.g., Markov chain Monte Carlo) to characterise the full space of possible solutions, due to their size and/or difficulty of realisation. Hence in order to maximise the diversity of solutions discovered, there is a need for a method of searching the space that informs the movement of the search as more information is gained about the space of solutions. One interesting problem space is the space of networks with a set level of degree distribution and global clustering,which is an important space for generating more effective null models.This space is combinatorially large, unknown for networks of any meaningful size (>1000 nodes), and complex [119, 127]. This is the focus of the first two chapters of this thesis (Chapters 2 and 3). In Chapter 2, we start by providing a MAP-Elites-style GA method for exploring valid networks using a subgraph-based encoding and mutation method. This method provides three main benefits to the search. First, encoding networks via their subgraph decomposition (i.e., the population count of each subgraph from a arbitrary family of subgraphs in the network) and realising them using CMA [151] provides scalability. Second, basing mutations directly on the higher-order structures of the networks improves the efficiency with which valid networks are found, whilst still providing more control at the level of higher-order structures. Finally, using a MAP-Elites-style grid-based storage of the generated population allows global novelty in the population to be

encouraged, without the need for comparisons with the full population; this therefore avoids the novelty cycling common to other novelty-driven methods [118]. This GA method is able to produce a population of valid networks that are more diverse than those produced using current state-of-the-art methods for valid network generation. However, due to the interconnected nature of this method of encoding (i.e., a change of one of the genes in the genome requires an accommodating change in one of the other genes to preserve the validity of the networks being produced), this GA method is very inefficient at finding valid networks. Additionally, the choice of the appropriate size/number of cells across the space of solutions requires some amount of prior knowledge of the diversity available in the space of solutions. Not only is this information not available in such an unknown search space, but a appropriate cell size also might not be homogeneous across the whole space. Further, the aim of this work is less focused on producing a large number of valid networks. We are not attempting a full coverage of the space of solutions, but rather mapping those areas of the space of most interest to the questions being asked about the space. As such, there is a need for further tailoring of the search focus so as to highlight these areas of most interest. These issues are the focus of our work in Chapter 3, in which we develop three additions to the method laid out in Chapter 2. First, we develop a Diophantine equation-based method of mutation, such that all mutations lead to valid networks. Second, we extend the standard MAP-Elites GA to allow for adaptive resizing of the cells across the space in response to how interesting a cell is throughout the search (referred to as the ARC MAP-Elites method henceforth). This means that an increasingly greater resolution of search is appled to interesting regions of the space. As part of this, we also develop the concept of "interestingness" in greater detail. Finally, we present a pairing of this adaptive resizing with the size of the mutations applied to the cells, such that the size of a mutation is proportional to the size of the cell it is stored in. This further focuses the search towards the smaller/more interesting cells within the space. We show that this ARC MAP-Elites + Diophantine mutation method is able to increase the diversity and

novelty of solutions found compared to the previous method.

Building on these results, Chapter 4 further expands our exploration of the the ARC MAP-Elites method in more general problem spaces. It provides the reader with a more detailed description of the relationship between different possible measures of interest (MoI) and the control variable touched on in Chapter 3, with examples for several distinct 2D landscapes. This chapter also provides sensitivity analysis on the range of the adaptive mutation size and its effect on the kind of population produced.

Finally, we benchmark ARC MAP-Elites against the standard MAP-Elites method in three problem landscapes: (1) a range of niching benchmark spaces; (2) the exploration of network structures (previously presented in Chapter 3); and (3) a hexapod robot walking simulation (a well-known example of the use of MAP-Elites in the development of multiple high-fitness solutions [29, 175]). In all landscapes, where known, we show that ARC MAP-Elites is able to identify more known niches in the space than MAP-Elites for a range of naive cell sizes.

We also showed that although there is a cost in the effectiveness of the overall optimisation of the search (as would be expected from any increasing divisions of the space), there is a significant increase in the diversity of results, both in terms of coverage of the space and secondary behaviours not included in the optimisation directly.

In Chapter 5, we summarise the main results of the research, highlight the limitations of our work, and propose further directions. We discuss the applicability of our work to the exploration of valid networks and how this work might be expanded to investigate other structural properties and/or behavioural dynamic diversity. Finally, we argue that our work has usefulness far beyond these specific search spaces, such as in improving our understanding of more black-box GA search spaces.

List of publications and author contributions

1. A genetic algorithm-based approach to mapping the diversity of networks sharing a given degree distribution and global clustering

Peter Overbury, Istvàn Z Kiss, Luc Berthouze (2016).
In International Workshop on Complex Networks and their Applications (pp. 223-233). Springer, Cham

- P. Overbury contributed toward conceiving the overall goals and design of the study as well as the use of a Map-Elite style genetic algorithm and produced the results.
- L Berthouze contributed toward conceiving the overall goals and design of the study as well as the use of CMA for encoding, the writing of the paper and the analyses of complex contagion results.

2. Mapping structural diversity in networks sharing a given degree distribution and global clustering: Adaptive resolution grid search evolution with Diophantine equation-based mutations

Peter Overbury, Istvàn Z Kiss, Luc Berthouze (2018).
In International Conference on Complex Networks and their Applications (pp. 718-730). Springer, Cham.

- P. Overbury contributed toward conceiving the overall goals and design of the study, the adaptive resolution grid search evolution method and produced the results.
- IZ Kiss contributed toward conceiving the overall goals and design of the study.

- L Berthouze contributed toward conceiving the overall goals and design of the study as well as to the writing of the pre-extended paper, the use of CMA encoding, the design of the diophantine equation-based mutations method and the analysis of complex contagion.

Chapter 2

Paper 1: A genetic algorithm-based approach to mapping the diversity of networks sharing a given degree distribution and global clustering

[1]

Peter Overbury[2], István Z. Kiss[3] and Luc Berthouze[2]

[2] Department of Informatics,
University of Sussex, Falmer, Brighton BN1 9QH, UK
[3] School of Mathematical and Physical Sciences, Department of Mathematics,
University of Sussex, Falmer, Brighton BN1 9QH, UK

[1]Appendix added to original publication, see section 2.6

2.1 Abstract

The structure of a network plays a key role in the outcome of dynamical processes operating on it. Two prevalent network descriptors are the degree distribution and the global clustering. While there are now effective and analytically-tractable mathematical models that can handle the degree distribution well, when clustering is also considered, most models will break down or only operate for networks constructed in particular ways, e.g., networks with non-overlapping triangles. Further complications arise from the fact that network-generating algorithms often induce changes in structural properties other than that controlled for. There is therefore value in getting greater understanding of the potential diversity of networks sharing a given degree distribution and global clustering. As the space of all possible networks is too large to be systematically explored, a heuristic approach is needed. In our genetic algorithm-based approach, networks are encoded by their subgraph counts from a chosen family of subgraphs. Coverage of the space of possible networks is then maximised by focusing the search through optimising the diversity of counts by the Map-Elite algorithm. We provide preliminary evidence of our approach's ability to sample from the space of possible networks more widely than some state of the art methods.

2.2 Introduction

Almost all complex systems can be modelled, to varying levels of detail, using networks whereby components of the system can be reduced down to nodes and to edges connecting them. Such an approach often makes it possible to pick out global behaviours dependent on the connections and/or relationships between different elements of the system that either would not have been noticed in isolation or could not be detected within large data sets [119]. The relationship between network structure and behaviour is the subject of much research in many areas such as epidemiology [32, 133, 70], social media [2] and

neuroscience [109]. Where analytically-tractable mathematical models are needed, two main network descriptors stand out: degree distribution and global clustering. Interestingly, while there are now effective and analytically-tractable mathematical models that can handle the degree distribution well [32, 133, 70], when clustering is also considered, most models will break down or only operate for networks constructed in particular ways, e.g., networks with non-overlapping triangles [180]. This sensitivity to how networks are constructed highlights the fact that, as shown by [59, 80, 82, 150] among others, many network-generating algorithms introduce changes in structural properties other than that controlled for, thus undermining both model accuracy and inference of any causal role for the properties of interest. How to create network *null models*, i.e., where the properties of interest are fixed and all other properties are sampled in an unbiased manner, is an open question. One major step towards realising such goal would be to get a greater understanding of the space of networks satisfying a given set of requirements, e.g., a given degree distribution and a given global clustering coefficient. For networks of non-trivial size, the space of all such networks is too large to be systematically explored and therefore a heuristic approach is needed. Our approach relies on two principles: (a) a parametrisation of networks in terms of subgraph decomposition, which significantly reduces the dimensionality of the encoding space when compared to the adjacency matrix as done in our previous work [129]; and (b) a search of the space driven by a process seeking to maximise the diversity of the networks being uncovered, thus biasing the exploration/exploitation trade-off toward exploration. The design and implementation of these two principles will be detailed in the following section.

2.3 Methods

2.3.1 Network encoding

A key challenge in exploring the space of networks satisfying constraints is that of network representation. In principle, the network's adjacency matrix would be a natural choice because it fully specifies the network. However, it suffers from two major drawbacks: scalability and unicity (two networks may have a distinct adjacency matrix but be isomorphic). Our previous work [129] using the adjacency matrix revealed an extremely wasteful process even for small sized networks ($N = 200$). The recently-proposed dk-decomposition [127] offers an attractive alternative through its use of joint degree distributions of different orders, however, as we will show, questions remain regarding the biased nature of the network generation process once the joint degree distributions have been set. Instead, building on our recent work [151], we propose to parameterise networks in terms of a (arbitrarily chosen[2]) family of subgraphs (see Figure 2.1 for a few examples). Concretely, we use the counts of each of the subgraphs in the family to yield an adjacency matrix using the cardinality-matching algorithm (CMA hereafter) [151]. CMA is a method inspired by the configuration model [77]. It assigns a set number of subgraphs of arbitrary structure in a network with a set degree sequence. Put simply, it works by assigning to nodes in the network hyperstubs of a certain degree as specified by each subgraph in the family. For example, triangles (subgraph C3) will require 3 hyperstubs of degree 2 whereas a Toast (see Figure 2.1 will involve 2 hyperstubs of degree 3 (corners with 3 edges) and 2 hyperstubs of degree 2 (corners with 2 edges). These hyperstubs are then selected at random and connected until there are no more left. When a new subgraph introduces self- or multi-edges, a new node is selected as in the matching algorithm [112]. When there is no option other than to add subgraphs over existing links or selecting multiple instances of the same node,

[2] see Chapter 3, paper 2, Section 3.3.1 for more information on this choice

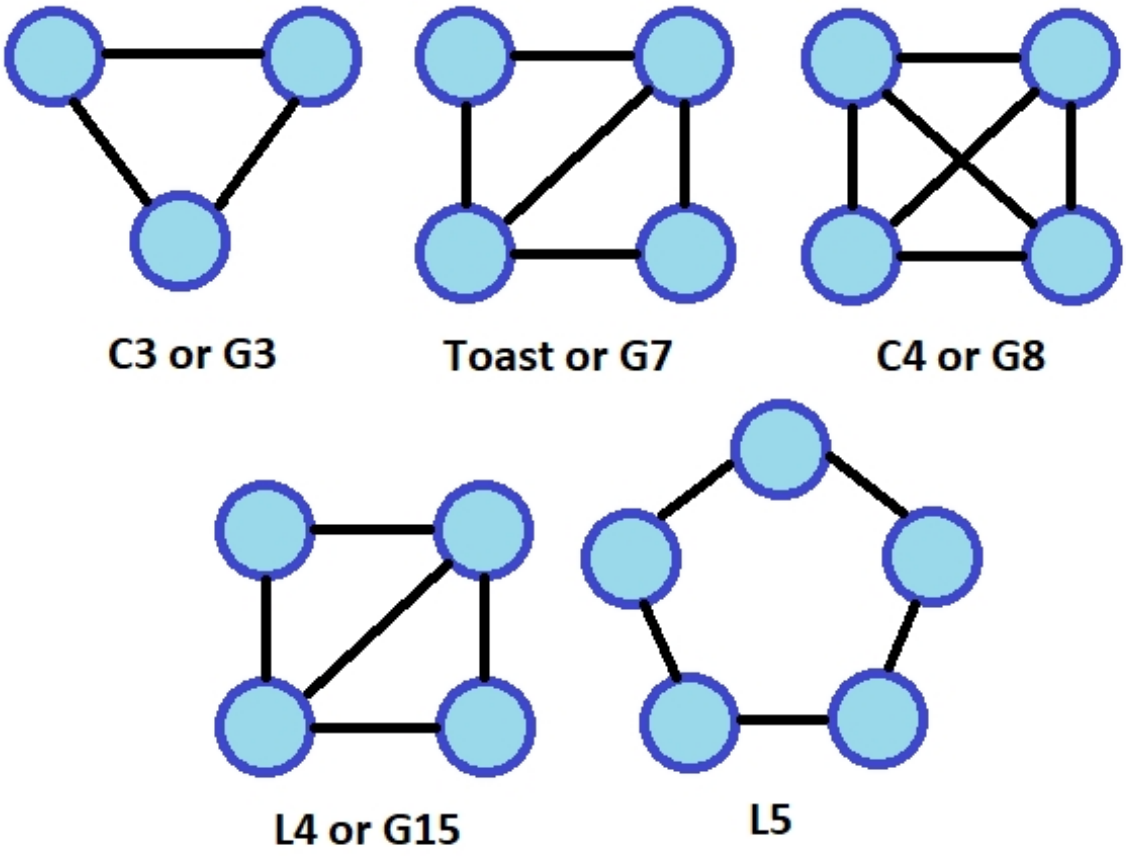


Figure 2.1: The set of subgraphs used to encode networks (single edges not included). Subgraphs in the top row will induce clustering in the network.

the process is restarted from scratch. To accelerate the process, in this work, only 80% of the networks' total edges were allocated to the specified subgraphs. The remaining edges were allocated as single edges to preserve the required degree sequence. Networks for which more than 20 nodes (out of a total of 1000) did not have the desired degree were excluded. Analysis of the networks produced (not reported here for reasons of space but available for an extended version, and see [151]) showed that the process still provides good control over most subgraphs, particularly (and advantageously in our context), those inducing clustering (i.e., C3, C4 and Toasts). Still, to avoid results being biased by a particular realisation, all measures reported in this paper were calculated by averaging over 5 network realisations. The reliability of the process is illustrated by Figure 2.2 which shows a compact spread of values of three network metrics (global clustering, mean shortest path length, mean betweenness centrality) for 10,000 realisations of a single network specification.
The choice of subgraphs is somewhat arbitrary and is a source of bias in itself. Here, we chose 3 subgraphs that induce clustering in the network (they are C3, C4 and toasts, see Figure 2.1). The other networks are loops that do not induce clustering. In this paper, only L4 and L5 were used. As a family, they provide flexibility and redundancy in the control for clustering. These 5 subgraphs have been shown in previous work to be those for which CMA showed most control over (as assessed by subgraph counting post realisation, see Chapter 3, paper 2 figure 2.2).

2.3.2 Exploration of the space of possible solutions

Our primary objective being an exploration of the diversity of networks preserving a given degree distribution and global clustering coefficient, our task can be thought of as a two-part optimisation: (a) of the features that must be shared by a network for it to be added to the population of valid networks and (b) of the diversity within this population of valid networks. Multi-objective optimisation is not a new problem and the more complex variant considered

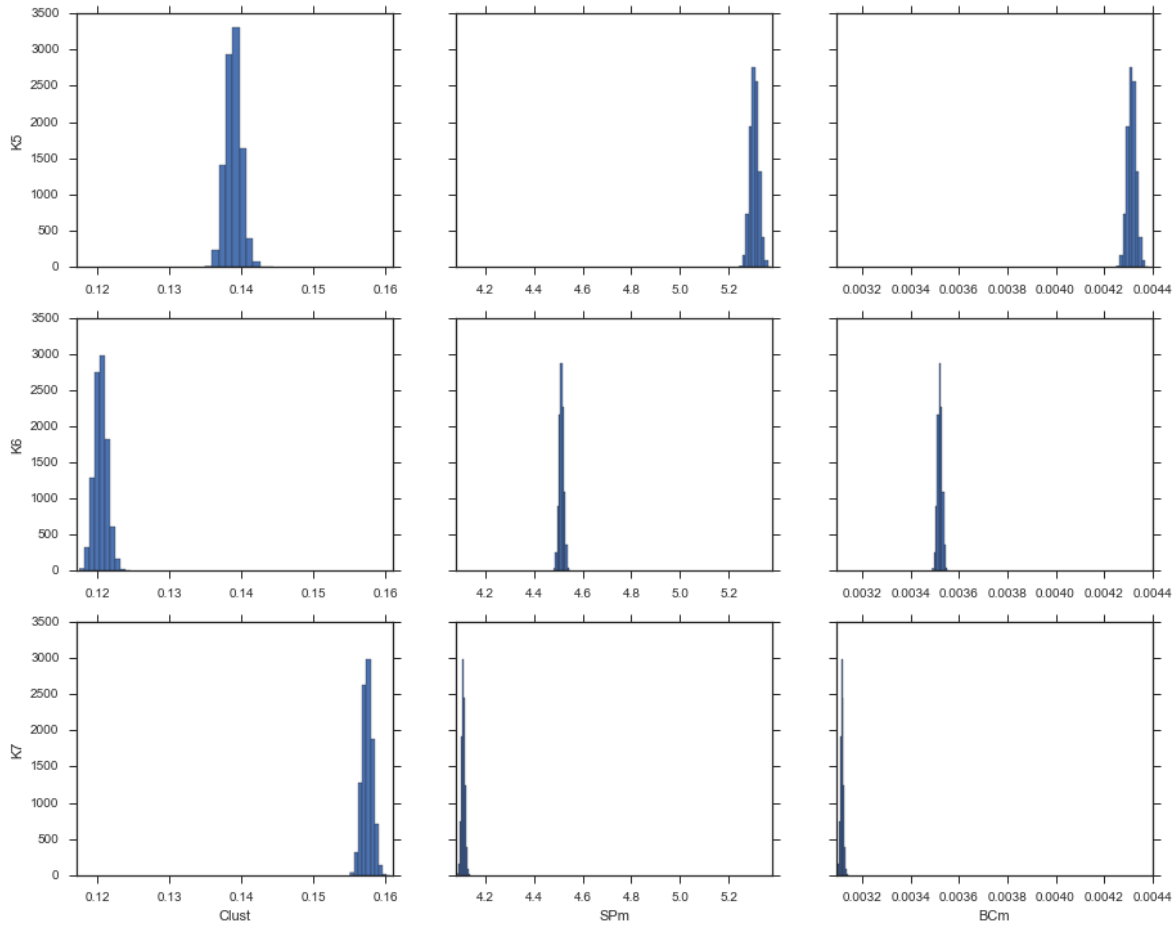


Figure 2.2: Histograms of global clustering (left column), mean shortest path length (middle column) and mean betweenness centrality (right column) for 10,000 CMA realisations of a single network specification with predicted global clustering of 0.14 ± 0.025. The top, middle and bottom rows correspond to regular networks with degree $K = 5$, $K = 6$ and $K = 7$ respectively.

here involving a changing measure of diversity within an actively changing population has recently been the focus of a number of methods in the field of genetic algorithms (GA) [92]. In their simplest form GAs work by taking a starting population of individuals, which are encoded so that each has a *genome* that represents the key features being studied, here, the subgraph counts. This population is then *evolved* through *genetic operations* that change the genome of individuals. This typically involves *mutations* – the adding or subtracting from parts of the genome – and *recombination* or *crossover* – the combining of two individuals into a new individual with a new genome. All individuals are then analysed for their *fitness* – the objective function in the optimisation process. Those with the lowest fitness are either removed, selected for genetic operations less often or both. This results in a population that, depending on the setting of the GA, moves along the search space towards areas of high fitness. An important implication is that the solutions are highly dependent on the choice of the fitness measure, the selective pressures used at each generation and the way that solutions are stored. Previous work based on the idea of optimising for diversity includes the generation of neural networks topologies for control of robots in which diversity of both behaviour and performance was optimised for [161] and our own work [129] in which we started exploring the feasibility of using GAs to optimise the diversity of networks satisfying structural constraints, albeit for small sized networks. The main limitation of these methods has been their focus on the optimisation of a few individuals to the best possible fitness over all their objectives (the Pareto front), often leading them to avoid equally valid/fit regions of the feature space. Here, we employ the recently proposed Multi-dimensional Archive of Phenotypic Elites (Map-Elite) method [118] which seeks to map the solution space through dividing the space into identically-sized multi-dimensional *cells* that cover a set range of values for each of the features used to describe the individuals. All individuals in the population are then placed in one of these cells and when new individuals are created they are assessed based only on individuals in that same part of the space. If there is no other in

the cell then the individual is deemed novel and is kept. If, instead, there is another individual already within the cell then only the individual with the greatest fitness is kept. This method allows for the promotion of novelty without comparison of the entire population whilst also optimising the fitness of the population.

2.4 Results

The experiments reported in this paper sought to map the diversity of networks of size $N = 1000$ satisfying the constraint of a homogeneous/regular degree distribution (with degree 5, 6 or 7 – as three distinct scenarios) and a global clustering coefficient of 0.14. Although our choice of network encoding is insensitive to network size, the CMA connection process is not. The size $N = 1000$ makes the experiments tractable, when deployed on the high performance computing facility. The three degrees considered enable us to assess the effectiveness of the method for networks with more ($K = 7$) or less ($K = 5$) flexibility in how to allocate subgraphs. For example, with $K = 5$, it would not be possible for a node to share a fully connected square ($C5$) and the degree 3 corner of a toast whereas with $K = 7$, the same node could accommodate that and an extra free edge. Our choice of global clustering coefficient is arbitrary although one should note that depending on the choice of subgraph family used to encode networks, some clustering values are more likely than others. With the proposed family of subgraphs and the relatively small degree, it would be difficult to generate highly clustered networks, and diversity would be extremely limited. A tolerance of ± 0.025 was used in evaluating the clustering fitness of networks. A tolerance is needed due to (a) the nature of the computation of the clustering coefficient and (b) the stochasticity in allocating subgraphs and any resulting byproducts [151]. This tolerance, which is reflected in the histograms of clustering values in Figure 2.2, corresponds to a maximum deviation of $\pm$ 8 triangles (subgraph C3) from the expected number of subgraphs

and is negligible given the number of triangles needed to achieve the required clustering.

2.4.1 Effectiveness of the mapping in terms of space coverage

To provide some quantitative assessment of the effectiveness of mapping, cells were configured for maximal resolution, meaning that all individuals within a cell would have the exact same subgraph counts. It should be noted at the outset (but this is currently the subject of further work) that starting out with maximal resolution is sub-optimal in terms of managing the evolutionary process. However, for the purpose of this assessment, it provides as detailed a picture as possible of the proportion of all possible encodings that is uncovered by the evolutionary process (with the caveat that with a limited number of generations, the actual number of cells uncovered can only be a tiny fraction of the total number of cells possible). In the following, when ignoring the fact that not all combinations of subgraph counts are actually realisable – graphicality of the network), the total number of cells possible is $1040625000000 = 333 \times 250 \times 250 \times 200 \times 250$ and corresponds to the product of the ranges of possible values taken by the counts of each subgraph in the family (this count is determined on the basis of the highest-degree hyperstub in relation to the total number of nodes available in the network). The actual total number of cells is found by subtracting from the above count those cells that correspond to non-graphical/non-realisable networks, namely, those where the total number of edges prescribed by the subgraph decomposition is above $(Nk)/2$ and where the number of triple hyperstubs from C4 and Toasts is greater than $(k/3)N$ – the maximum number of triple hyper stubs allowed by CMA in a network. Coverage of the space at various points during the process is shown in Table 2.1. Given the maximum resolution and the fact that each generation only produces one new network, the actual percentage of coverage is very small. However, the table shows two important results: (a) the rate at which new cells are explored in relation to the number of generations is almost 1 suggesting that cells are not revisited (this would no longer be the

K	21,000 gen		42,000 gen		63,000 gen	
	Explored	Valid	Explored	Valid	Explored	Valid
5	20783	12995	41546	25952	62286	38852
6	20824	18266	41583	36596	62349	55009
7	20845	18691	40646	36680	62431	56435

Table 2.1: Number of explored and valid cells uncovered by the evolutionary process at various time points for the three scenarios ($K = 5, 6, 7$) considered. In all cases, networks have size $N = 1000$ and the family of subgraph considered is (C3, C4, Toast, L4 and L5) with a desired global clustering of 0.14 ± 0.025. For reference, the total number of cells possible (after removal of non-graphical solutions) is $\sim 10^{12}$. Each generation can produce at most one new network.

case if cells had lower resolution); (b) the rate at which valid networks are produced is roughly constant as the number of generations increases.

Importantly, we note that this table does not provide any information regarding coverage of the space of valid networks, those with correct degree distribution and global clustering within ± 0.025 of the desired clustering. Whilst the search is focused on finding valid cells (rather than all possible cells), we do not have any estimate for the total number of possible valid networks in the space of all possible networks. Figure 2.3 provides a different perspective on this by using low-dimensional projections of the space of networks explored and valid. Where possible, non-graphical solutions have been highlighted. The Figure reveals that despite the limited number of generations (again, corresponding to a very small percentage of all possible configurations) there is evidence of fairly uniform sampling as far as explored cells are concerned. The Figure further reveals pair-wise relationships between counts of subgraphs that reflect the constraints of the problem. For example, when two clustering-inducing subgraphs are considered (e.g., C4 and Toast) there is a distinct relationship whereby configurations with larger numbers of C4s have smaller numbers of

Toast and conversely. Instead when clustering-inducing subgraphs and non clustering-inducing subgraphs are considered (e.g., C3 and L4) valid configurations can be found throughout the space of explored solutions. Areas that are not explored are typically reflecting configurations for which although no graphicality condition is being violated as far as the particular pair of subgraphs is concerned, no network realisation is possible when taking into account the other dimensions.

2.4.2 Comparison with other methods

Whilst the above results point to evidence of diversity in terms of subgraphs a more useful basis for evaluating the effectiveness of our approach is to assess the extent to which networks uncovered show greater diversity than can be expected from methods currently available to generate networks satisfying the same constraints. Since subgraphs counts are explicitly controlled by the evolutionary process, they would not be a fair metric for comparison. Instead, we considered two global structural properties: mean shortest path length and mean betweenness centrality (BCm) – although as both show a high degree of correlation, only betweenness centrality will be reported below. These properties are important determinants of behaviour in networks [120]. Two state of the art network generating methods have been used for this comparison: dk-series decomposition [127] and BigV rewiring [67]. For the former, we used dk2.1 which preserves degree distribution and global clustering (dk2.5 would also preserve local clustering which is overly specific for our purpose). Since the dk method requires a seed network to operate, one network was chosen at random among those generated by our approach. For the latter, the rewiring algorithm was applied to a single random network with homogeneous degree distribution who was rewired until desired clustering was achieved (with a maximum of 40000 rewirings). For both BigV rewiring and dk decomposition, the number of networks generated was set to the number of networks produced by the GA. Figure 2.4 reveals that the range of mean

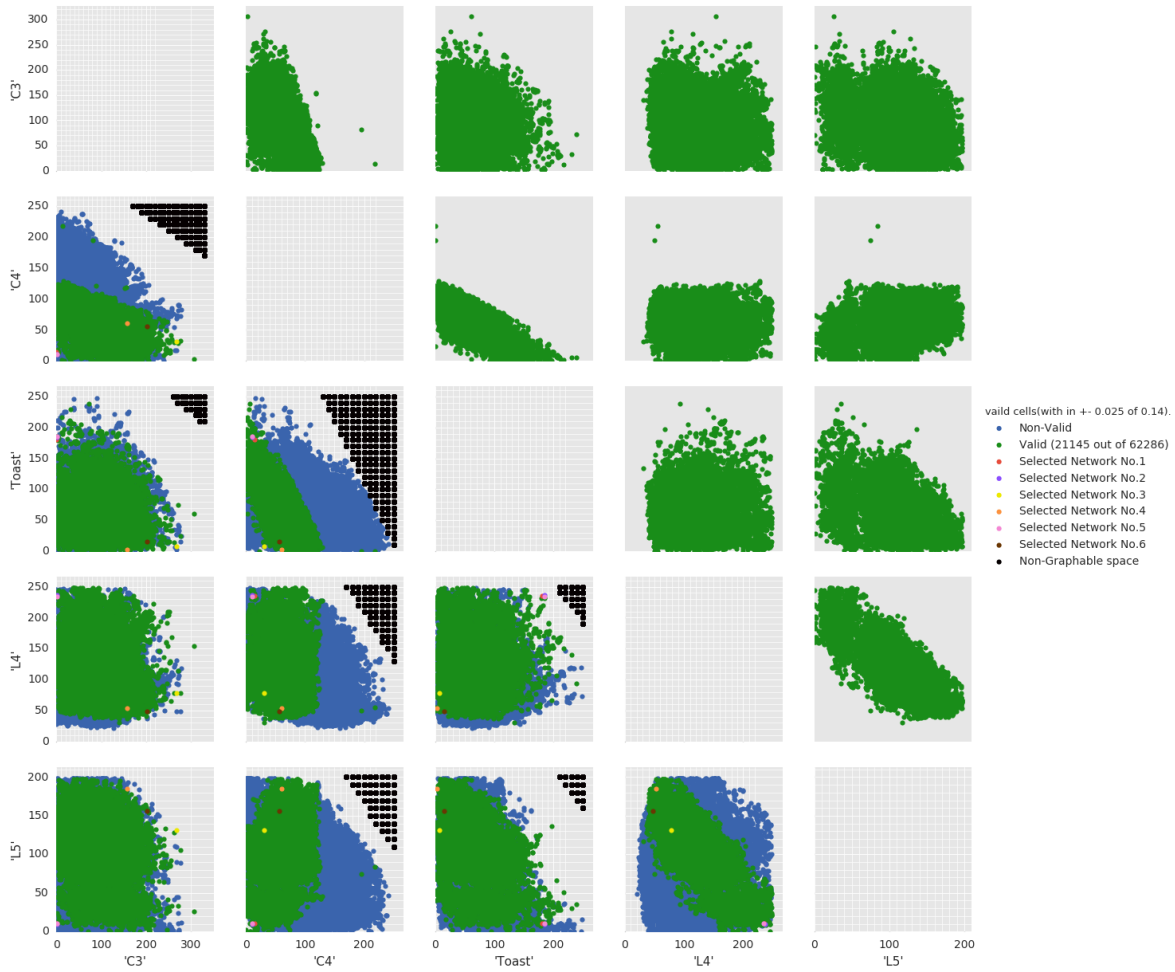


Figure 2.3: Low-dimensional projections of the configurations discovered by the evolutionary process (both those that were explored but not necessarily satisfying the constraints – in blue – and those that were valid – in green) after 63040 generations. Each dot denotes a network whose coordinates are the counts for the subgraphs shown in the horizontal and vertical axes. A dot does not define a unique network, however, as the projection can mask great diversity in the remaining 3 dimensions. Six individual configurations are identified which will be discussed in Section 2.4.3.

betweenness centrality for networks produced by our approach is greater than that of either (or even both of) the dk- and BigV-produced networks, suggesting that a wider area of the space of solutions was explored. This holds for all three scenarios ($K = 5, 6, 7$). An important correlate of this finding is that neither BigV rewiring nor dk-decomposition can claim to generate null models. Interestingly, the networks produced by both methods do not appear to overlap suggesting that either methods generate networks in different areas of the space of solutions. Likewise, although our method appears to sample more widely than BigV rewiring and dk, full overlap only occurs for $K = 7$ whereas there is almost no overlap for $K = 5$. It remains to be seen whether, given more time, our method would uncover these areas of the space of solutions. Finally, given that the dk networks were produced from a single seed, it is worth pointing out that there was no obvious correlation between the betweenness centrality of the seed and the mean betweenness centrality for the dk-generated networks. The extent to which the choice of seed conditions the distribution of networks generated remains unclear.

2.4.3 Impact of diversity on behaviour

A fundamental observation underlying this work is that higher-order structure matters, see [150] for example. Here, we illustrate this by selecting 3 pairs of networks and assessing the impact of their differences by simulating dynamics on them. A first pair of networks (A and B) maximised diversity in terms of the subgraph counts. The second pair (minC4 and maxC4) maximised the difference in the number of C4 subgraphs (fully connected squares) involved in the network. The final pair (BCmMin, BCmMax) maximised the difference in mean betweenness centrality. Two classical dynamics were tested: SIR (susceptible-infected-recovered) and complex contagion. In an SIR epidemic, a susceptible node connected to an infected node becomes infected at rate τ, and once infected it recovers at rate γ, independently of the network. All processes are considered to be independent

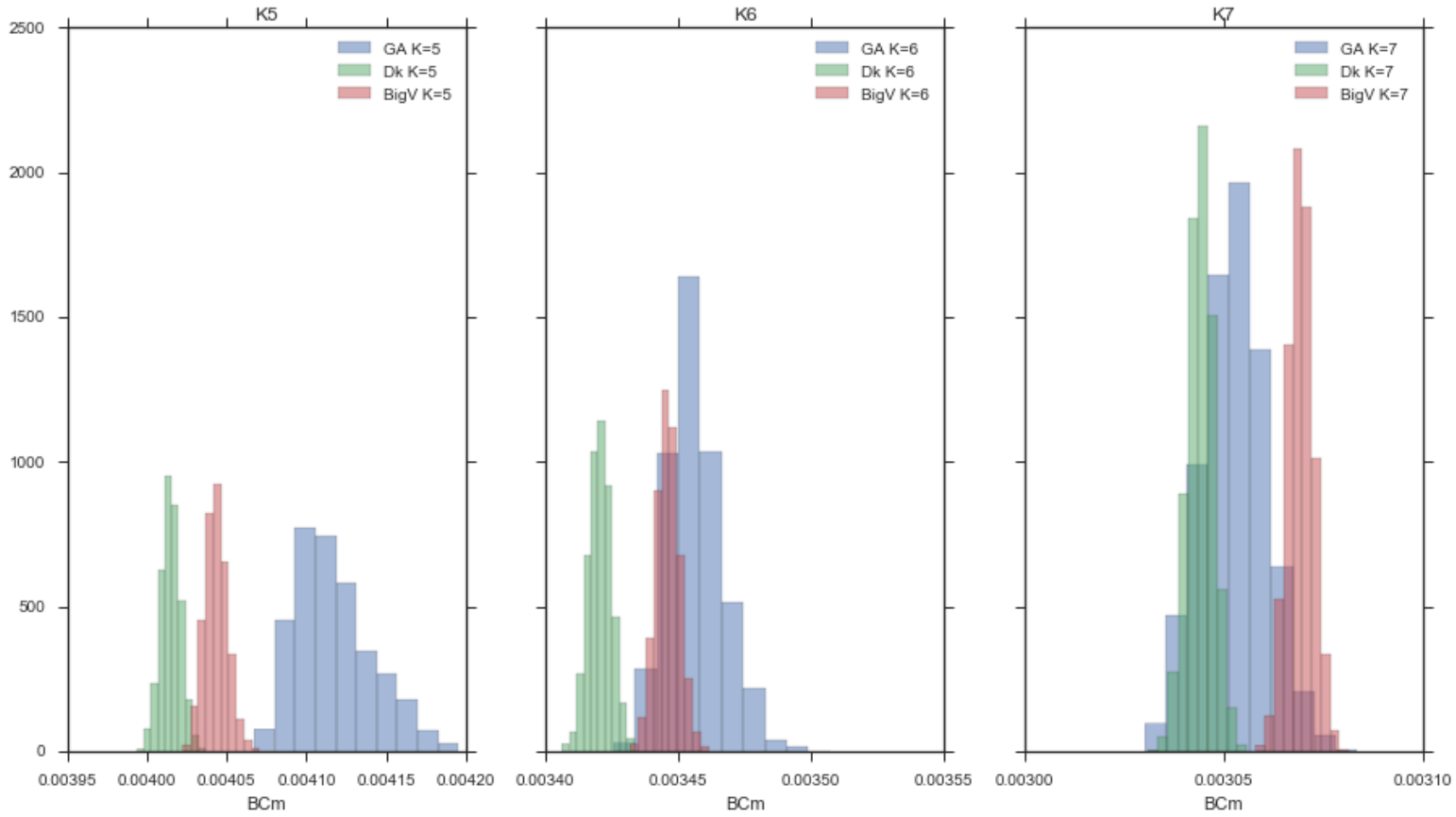


Figure 2.4: Histograms of the mean betweenness centrality for the proposed method (blue), BigV rewiring (red) and dk2.1 (green) for each of the three scenarios: $K = 5$ (left), $K = 6$ (middle), $K = 7$ (right). The same number of networks was used for all three methods.

Poisson processes. The final epidemic size is computed by counting the number of recovered individuals at the end of the epidemic when no further infected nodes remain. Each individual epidemic starts with one single infected node. None of the pairs showed any meaningful difference (results not shown but available for an extended version). The complex contagion model differs from the above by requiring that susceptible nodes are exposed to multiple infectious events before becoming infected. Further, these events must be from different infectious neighbours as only the first infection attempt from an infectious node counts; and infected individuals remain infected for the duration of the epidemic. This dynamics is known to exhibit a critical transition in relation to the number of infected nodes at the outset of the epidemic. Here, we identified the critical transition through tracking the mean and standard deviation of two quantities – final size and time to reach final size – when systematically varying the number of initial seeds. Figure 2.5 shows that all three pairs of networks show distinct profiles in both quantities when the number of initial seeds vary between 1% and 20%. Using maximal variability in both the time needed to reach final size and final size as marker of the critical transition, the figure shows that the parameter value at which the critical transition occurs differs substantially between networks (see distinct peaks).

2.5 Discussion

In this paper, we have proposed a new GA-based approach to generating networks preserving degree distribution and global clustering. Our approach is focused on maximising the diversity of the networks being created. Since it is impossible to quantify the extent to which the entire space of solutions has been sampled, we have provided evidence of the effectiveness of the method by comparing it to two state of the art network-generating methods, dk-series decomposition and BigV rewiring and showing that

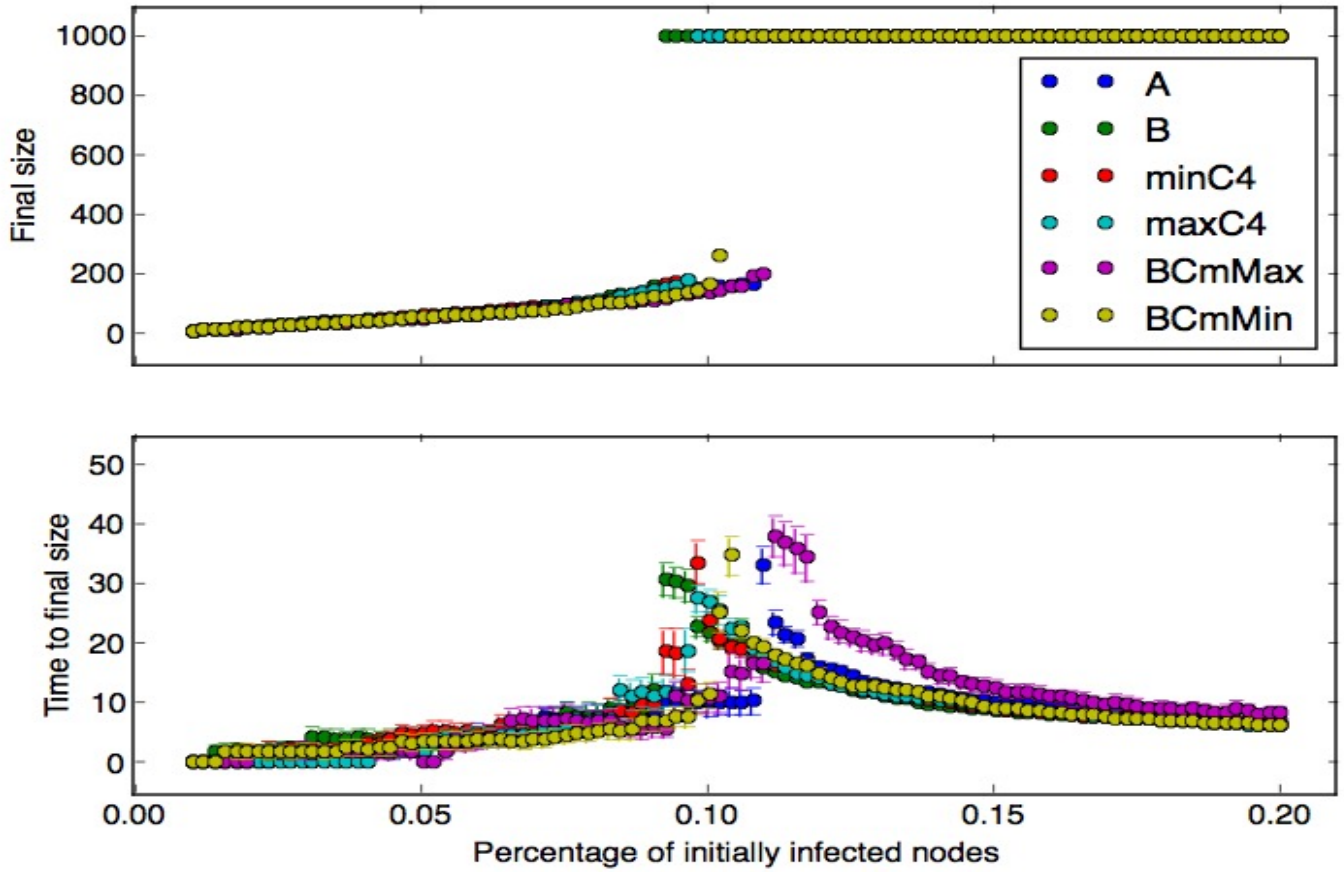


Figure 2.5: Impact of differences in higher-order structure on the critical transition for complex contagion simulations on 3 pairs of networks maximising diversity in terms of subgraph decomposition (A-B), involvement of C4 subgraphs (minC4, maxC4) and mean betweenness centrality (BCmMin, BCmMax). See text for details.

our method generates more diversity. Whereas coverage of the space of solutions using our method will depend on the number of generations available, both BigV rewiring and dk-series decomposition depend on a mixing time being reached. Care must therefore be taken in making definite statements about the ability of these methods to sample the range of networks found by our approach. However, given the same number of steps, there was greater diversity using our approach. This provides evidence for the usefulness of our method in the evaluation of the level of bias shown by current network generation methods. Much further work is needed to strengthen our framework, especially given that it is itself subject to a number of biases. For example, whilst encoding in terms of subgraphs provides much flexibility and scalability, it is itself a source of biases. At this time, it is unclear how a different choice of family would affect the diversity of networks uncovered. On the bright side, we believe that our starting scenario of networks with homogeneous distribution and low degree actually made it much harder to find diversity in the networks. The immediate focus will be to consider heterogeneous distributions with higher degrees. Whilst it will not affect computation time, it will provide much more flexibility for the network connection process (CMA) to realise networks (as well as remove the need to allow for 20% free edges, thus providing further control).

2.6 Appendices

Parameters of Evolutionary Algorithms

Chance of mutation per subgraph =50%

Mutation Rate= random number between range of +0.1 to -0.1

Number of individual evaluated per generation =1

Chapter 3

Paper 2: Mapping structural diversity in networks sharing a given degree distribution and global clustering: adaptive resolution grid search evolution with Diophantine equation-based mutations

[1]

Peter Overbury[2], István Z. Kiss[3] and Luc Berthouze[2]

[2] Department of Informatics,

University of Sussex, Falmer, Brighton, BN1 9QH, UK

[3] School of Mathematical and Physical Sciences, Department of Mathematics,

University of Sussex, Falmer, Brighton, BN1 9QH, UK

[1] As extended from the published paper, from 8 pages to its current length

3.1 Abstract

Methods that generate networks sharing a given degree distribution and global clustering can induce changes in structural properties other than those which are controlled for. Diversity in structural properties, in turn, can affect the outcomes of dynamical processes operating on those networks. Since exhaustive sampling is not possible, we propose a novel evolutionary framework for mapping this structural diversity. The three main features of this framework are: (a) subgraph-based encoding of networks; (b) exact mutations based on solving systems of Diophantine equations; and (c) a heuristic diversity-driven mechanism to drive resolution changes in the MAP-Elites algorithm. We show that our framework can elicit networks with diversity in their higher-order structure and that this diversity affects the behaviour of the complex contagion model. Through a comparison with state of the art clustered network generation methods, we demonstrate that our approach can uncover a comparably diverse range of networks without the need for computationally unfeasible mixing times. Further, we suggest that the subgraph-based encoding provides greater confidence in the diversity of higher-order network structure for low numbers of samples and is the basis for explaining our results with the complex contagion model. We believe that this framework could be applied to other complex landscapes that cannot practically be mapped via exhaustive sampling.

3.2 Introduction

Almost any complex system involving the interaction of constituent components can be represented as a network, with networks becoming a paradigm of choice for modelling and analysing such systems. It is now well known that node-level and structural properties of networks (e.g., degree-distribution, assortativity, clustering, or modularity) can fundamentally affect the way the system operates [120, 146, 133, 70]. Clustering, in particular, has been the subject of much work, leading to both empirical and analytical

results [182, 42, 59]. However, there is also a growing awareness that local structure (e.g., subgraph composition) may also have an important impact on dynamics [66, 77, 151].

To enable a more compelling demonstration of this, methods that can sample the space of networks satisfying set constraints (e.g., degree sequence, assortativity, or global clustering) are required. Currently, the available network generative methods can be categorised in terms of where they fall within the 'one-shot' to 'growing/developmental' spectrum (see [12] for a more comprehensive treatment of this topic). Algorithms on the 'one-shot' end of the spectrum produce a single network. One of the most popular examples of such a generative model is the exponential random graphs model [153], in which it is assumed that links are random variables and that each realisation comes from a probability distribution of graphs from a given number of nodes.

The probability of observing any particular graph g is usually of the form $\frac{1}{K}\exp(\sum_{i,j}\eta_{ij}y_{ij})$, where: y_{ij} is a realisation of a random variable Y_{ij}, which determines whether nodes i and j are connected; η_{ij} are parameters used to tune the importance of various links that are set during construction; and finally, K is simply a normalising constant.

Variants of such models providing more control over the relationships between nodes include graphons [55, 101]. These are defined by a symmetric measurable function $W : [0,1]^2 \rightarrow [0,1]$, where each node j of the network is assigned an independent random value $u_j \sim U[0,1]$ with edges (i,j) independently included in the graph with probability $W(u_i, u_j)$.

At the other end of the spectrum, one can find methods that involve fundamentally rewiring existing networks, which are typically based on Markov chain Monte Carlo processes [35]. To construct networks with a fixed degree sequence and global clustering coefficient, BigV [9, 59, 67] starts from a random network and performs a series of degree-preserving rewiring operations which increase clustering. The process is repeated until the desired clustering coefficient is achieved. This process yields one network and must be repeated to

generate a population of networks. Conversely, dk-series decomposition [127] uses rewiring to generate randomised versions of a given network that preserve network characteristics from the average degree (dk = 0) to the global clustering coefficient (dk = 2.1). Furthermore, the dk-series is able to deal with any set degree distribution that is realisable and even to control for local clustering with dk = 2.5. In principle, rewiring approaches could be used to sample the network space; however, they do not actually provide any control over which local higher-order structure property is being changed. Furthermore, the question remains of whether (even given sufficient time) these approaches will necessarily cover the full range of possible networks, depending on the seed network [127]. Two common features of both approaches are that: (a) there is great computational cost to mapping the network space; and (b) they do not lend themselves well to controlling/assessing the make-up of networks beyond the specified characteristics.

An alternative approach is therefore to not attempt to be as exhaustive as possible with our search, but rather to maximise the diversity of networks found within a given amount of time. In the kind of scenario we consider, population-based algorithms (see [54, 187] for reviews, and also [99]) can prove particularly helpful in these massive unknown problem spaces. More specifically, there has been a growing body of research into quality diversity (or illuminative) algorithms, whose focus is to discover both diverse and high quality solutions at the same time (see [141, 30, 7, 142, 140]) by favouring exploration over exploitation of the space. In the Multi-dimensional Archive of Phenotypic Elites (MAP-Elites) approach [118], the space of features (or behaviours) an individual might possess is divided into cells that act as niches to the population, forcing new individuals to only compete with individuals in the same cell. As a result, only the fittest (the elite) in each cell remains in the population, thus providing a collection of diverse, high-performing individuals. In previous work [130] (see Chapter 2), we combined the MAP-Elites method with the cardinality matching algorithm (CMA) [151]. CMA breaks the problem of generating networks down

to the subgraph level (subgraphs being small structures) by arranging set populations of these subgraphs in such a way as to satisfy a prescribed degree distribution and global clustering coefficient. Although our method elicited a wide range of diversity at both structural and behaviour levels compared with other methods of network generation, it suffered from being very slow and ineffective at producing large pools of networks.
We substantially improved upon this method by introducing two major changes:

1. an exact Diophantine equation-based *mutation* method which guarantees that all individuals in the population are fit (i.e., they satisfy the constraints);

2. an adaptive resolution mechanism, whereby the size of a niche changes during evolution in response to the level of variation between individuals. This allows us to efficiently control the trade-off between coverage and diversity.

In this paper we show the effectiveness of this method for exploring diversity in network structures that only share set clustering and degree distribution.

3.3 Methods

3.3.1 Defining the search space: network encoding

When examining many real world networks, it has been noted that they often contain a statistically surprising number of triangles, sets of three vertices connected by three edges [182, 144]. These small structures appearing to a statistically high degree are know as motifs and are often thought to have a impact on the functions of some networks [19, 162]. Similarly, other small structures, hereafter referred to by the general term for any set structure, "subgraphs", have been noted within a wide range of networks to perform modular tasks that contribute to the overall operation of the network [113, 154]. Despite this, many of the current methods of network generation do not control for changes in these higher-order structures. Thus, in order to provide control and to explicitly manipulate the

higher-order structures whilst preserving a set degree sequence and global clustering coefficient, we encode our networks in terms of the population counts of each subgraph from an arbitrarily chosen family of subgraphs (e.g., {△, □, ◪, ⊠, ⬠}), as was done in [130] and [131]; we call this our "genome" for each network generated in the search. The subgraphs chosen for this encoding should include at least one clustering-inducing subgraph (i.e., involve at least one set of three nodes connected by three edges) in order to have control over global clustering of the generated networks. Other than this constraint, the subgraphs used do not have to be of any particular size or number of edges, but smaller subgraphs (i.e., those involving fewer nodes) will of course imply a tighter control over the generated networks and will allow a greater range of arrangements in the realised network.

Furthermore, we avoid using subgraphs with loose edges (i.e., where one or more nodes are only connected via one edge), as these have a higher chance of creating unintended subgraphs during the allocation process (i.e., ⊏ is very likely to become □ when all subgraphs are connected in the network, lowering the control any encoding has over the generated network structures). The same is true for single edge connections (|), as changing the number of these is akin to adjusting the level of control of "free edges" that a particular encoding has over the CMA realisation, as described later in this section. Here we have chosen to use a genome of five subgraphs consisting of three clustering-inducing subgraphs (△, ◪, and ⊠) and two non-clustering-inducing subgraphs (□, and ⬠). These subgraphs were chosen for their small size, with a △ being the smallest subgraph of significance as described above, and each of the others being as small as possible without repetition (i.e., the ◪ is two △ sharing a edge, and the ⊠ is the maximum packing of △ possible with four nodes). In order to generate networks from this genome, i.e., using the counts of each subgraph to generate a network of the the correct degree distribution (also called realisation), we use the CMA [151]. To achieve this, the CMA takes a subgraph sequence denoting the number of subgraphs of each type involved with each of the N nodes

in the network. It uses this to allocate subgraphs to nodes in the network so that they fit with a given degree distribution. With the homogeneous networks explored here, in which the degree of all nodes is equal to value K, the networks are we generated by sampling N times from a random binomial distribution with a maximum range of 0 to F_i, where F_i is the maximum number of whole subgraph i that can fit around nodes of degree K, and a probability of $\sum_i \left(Po_i/c_i\right)/N$ where Po_i is the population count given from the genome for subgraph i and c_i is the number of nodes needed to make subgraph i. It is important to also stress at the outset that the algorithm is not exact: first, to mitigate the combinatorial complexity of satisfying all constraints, it is necessary to specify a fraction of edges not accounted for by the subgraphs (free edges) (see Section 2.3.1 of Chapter 2); second, as described in [151], the allocation process can lead to by-products (particularly when free edges are involved). For example, the addition of a free edge can lead to two distinct △ turning into one ◹ and one △. Figure 3.1 illustrates this problem by showing that whilst CMA yields fairly good control over the ⊠, there is more uncertainty for △ and ◹ (note that by-products of non clustering-inducing subgraphs are not an issue, since they do not have any impact over the clustering coefficient). Nevertheless, the right-hand side panel in Figure 3.1 demonstrates that despite the by-products, the process yields acceptable control over the global clustering coefficient. The clustering values obtained never exceed the target value by more than 0.003 (i.e., at most 21 △) in the 1,000 regular networks of size $N =$ 1,000 and degree $k = 7$ tested. Nevertheless, this control over the clustering will be affected by the generation of an increasing number of edges in the network (i.e., there will be a greater number of uncontrolled triangles created for networks with more available edges than in networks with fewer edges). Thus, to account for this effect and to avoid differences in the quality of results with varying levels of network density (numbers of edges), we set the tolerance of clustering on the networks generated to be within a range of +0.005 of the clustering value set. To further improve the speed and reliability of the method, we set an

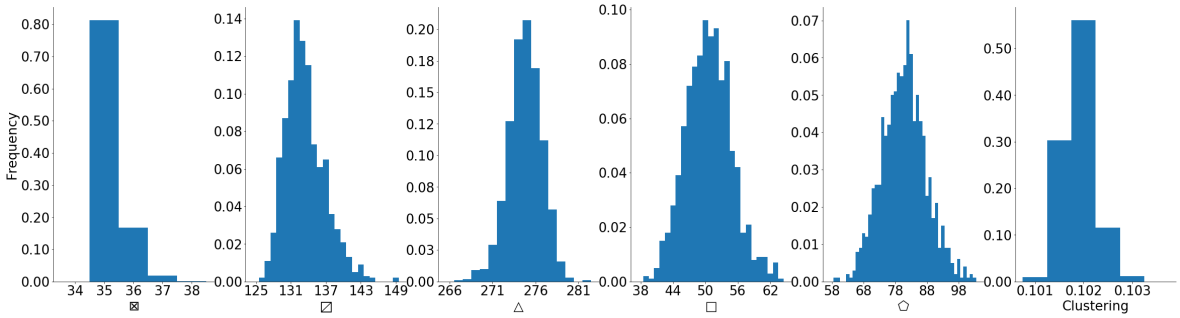


Figure 3.1: First five panels: Histograms of the prevalence of each subgraph for 1,000 networks generated by the CMA for specification (⊠:35, ◪:128, △:277, □:35, ⬠:42). The subgraphs were counted using the method in [149]. The number of △ denotes the number of △ not involved in any other clustering-inducing subgraphs. Last panel: Histogram of global clustering coefficient values [32] for the 1,000 networks considered. The target global clustering coefficient was 0.1. The excess clustering seen here is the result of the creation of unintended clustering-inducing subgraphs, as discussed in Section 3.3.1 above.

upper limit on the subgraph population counts for each network that avoids counts that will never lead to "valid" networks (i.e., subgraph counts that are not graphical [102], or that will not have the correct degree distribution and global clustering). In previous work [130], we used a very conservative boundary to the genomes, by only allowing a maximum number of subgraphs that could fit without sharing any nodes (i.e., N/Ci, where N is the number of nodes in the network, and C_i is the number of nodes involved in subgraph i). Here we improve upon this by allowing each of the nodes allocated to a subgraph to be involved in more than one of the given subgraphs, resulting in a much larger boundary of the genome space. This boundary is defined by $\sum_1 \left(N/C_i\right) \times F_i$ where F_i is the maximum number of whole subgraphs i that can fit around node of degree K (i.e., for ◪, in which half of the nodes involved will have degree 2 and half will have degree 3, $F_i = 2$ for $k = 7$); thus, for a network of $N = 1{,}000$ the maximum of ◪ we would allow would be 500.

3.3.2 Defining movement within the search space: exact mutations

A standard part of almost all evolutionary algorithms involves the alteration of genomes into new individuals, referred to as a "genetic operation" [114]. This normally consists of

mutations in which the genome of one individual is altered to create new offspring and/or crossovers in which the genomes of multiple individuals are combined in some way to create new offspring. New solutions are created via these mechanisms; thus, their implementation has a large effect on how the search progresses around the space of possible solutions, with the size and method of the implementation possibly leading to uneven coverage of the space of solutions [18].

Here in this work, we have a complex problem space in which not all combinations of subgraphs will be graphical (i.e., realisable as a connected network), or even create networks with the target structural features (i.e., degree distribution and global clustering). In previous work [130], we performed mutations to the genome of subgraph counts, such that each subgraph in the selected genome had a 60% chance of being changed by $\pm$2–50 subgraphs of that type, which represents a very frequent addition/removal of a relatively small number of subgraphs. This method was chosen to emphasise the exploration of new possible subgraph combinations; however, it did mean that non-valid networks were found very often, with the majority of mutations resulting in non-graphical networks. Furthermore, as these mutations did not preserve the relationship between the different subgraphs in a genome, even when the correct global clustering was achieved via optimisation, the validity of the genome coding could easily be lost in the offspring. As such, 38% of K5, 12% of K6, and 10% of K7 mutations resulted in networks that did not share the target global clustering. Both of these factors made the method far too slow at generating valid networks for practical use, with 63,000 generations only resulting in the following numbers of networks sharing a global clustering of 0.14: 38,852 networks for K5; 55,009 for K6; and 56,435 for K7.

Thus, there is a need for a genetic operation that allows solutions to be avoided that will not result in either graphical networks or networks that do not maintain the global properties being targeted (i.e., degree distribution and global clustering), whilst still allowing for full coverage of the space of possible subgraph configurations (i.e., without restricting the areas

the search is able to explore). This is not a trivial task however, as interdependence within a valid genome (as stated above) means that a change to one subgraph count will require changes to one or more of the other subgraphs count. For example, a constant global clustering coefficient requires that the addition of one ◿ comes at the loss of two △. However, simply reducing the number of △ by two does not suffice because such a operation would leave a deficit of one edge at the network level. In this paper, we cast the problem of identifying degree- and clustering-preserving mutations ("exact mutations", thereafter) in terms of solving a Diophantine problem (i.e., finding the integer solutions to an undetermined system of linear equations). Formally, an exact mutation is an integer solution of the system $A\mathbf{x} = b$, where: $\mathbf{x}$ is a column vector of n rows and specifies the change in the number of each of the subgraphs specifying the network (i.e., n is the cardinality of the family of subgraphs used to parameterise the networks; $n = 7$ throughout the paper); and A and b are a $3 \times n$ matrix and a column vector of 3 rows, specifying the three constraints that a mutation $\mathbf{x}$ must satisfy. These constraints are: (i) the change in the total number of triangles in the network must be 0; (ii) the change in the total number of edges in the network must be 0; (iii) the size of the change for the subgraph count(s) being mutated has the required size (see below). Note that the third constraint is purely for programming convenience, as only the first two rows specify constraints between subgraphs. To illustrate the principle, given individual (⊠:61, △:283, □:110, ⬠:142, ◿:87) and a required mutation of size 2 in the number of ◿, a possible vector $\mathbf{x}$ is (⊠:-1,△:0,□:-1,⬠:0,◿:2) leading to the new network specification (⊠:60, △:283, □:109, ⬠:142, ◿:89). It is easily verified that the gain of four △ via the addition of two ◿ is compensated by the loss of one ⊠, whereas the resulting excess of four edges ($2 \times 5 - 1 \times 6$) is absorbed by the loss of one □.

Solving an underdetermined system of Diophantine equations in general is a hard problem; however, finding solutions with the lowest Euclidean norm is easier [61]. To accomplish this, we used the following implementation: http://github.com/tclose/Diophantine (last

accessed 23/2/2020). The implications of this is that solutions tend to be homogeneous (with little difference between the absolute values of the components of the solution), which significantly biases the space of possible mutations available. For this reason, we built a catalogue of solutions by systematically enforcing values for each component of the solution vector. In the experiments that follow, the catalogue of possible mutations for 5, 7, and 9 homogeneous networks comprised 581 exact mutations for each of these three conditions, with mutation sizes ranging from 2 to 128 and involving from 1 to 3 subgraphs in any single mutation. This means a maximum mutation size of two times the maximum resolution of a cell size of 64 (which is discussed in more detail later in Section 3.3.3) and a minimum mutation size of two, which is the smallest change in any one subgraph used here that would have a significant effect on the structure of networks of this size and density (i.e. it is the smallest even number, and is discussed later in this section). To be clear, given that clustering is just preserved from the parent network, the catalogue of possible mutations is only affected by the total number of edges required to be kept constant. Thus, it can be easily adapted to any degree distribution, as long at the total number of edges is provided. There are three additional observations to be made. First, because all computations are on integers, then given a particular family of subgraphs, some mutations are not possible (i.e., the solver returns no solutions). A trivial example of this is that, given a family in which the only clustering-inducing subgraphs are $\triangle$, $\boxtimes$, and $\boxslash$, it is not possible to mutate the number of triangles by an odd number. Second, even if there is a solution, there is no guarantee that the network thus specified will be graphical or realisable (in a configuration model sense). In our implementation, we leave it to the CMA algorithm to make this determination. Finally, due to the need to control for both clustering and the degree distribution (here controlled via the total number of edges provided in the genome and the degree sequence provide to the CMA), there is a limit on the level of clustering that can be achieved via this method of mutation. This is because clustering above the level that can be gained without sharing of edges would

require triangles sharing edges in the network to be counted; thus, this is not possible with this method of edge control (see Figure 3.2). In this paper, this is not a significant disadvantage, as the method of network realisation, CMA, shares the same issues [151]. As such, higher levels of clustering could therefore be explored using this framework by treating the total number of edges in the network as a goal of optimisation, which is increased toward the target level needed, and having the mutation above deal only with maintaining a set level of global clustering. However, this would require an alternative method of network realisation, as the CMA method used here also would share this limitation.

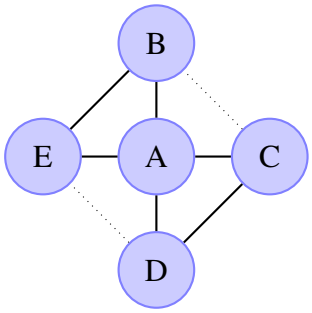


Figure 3.2: Demonstration of the problem of increasing clustering without using subgraphs involving shared edges. We can see that with the current two △ (shown with solid black lines), the only way to increase the number of △, and thus the level of clustering, would be to add one or both of the edges shown by the dotted lines. This forces the requirement for subgraphs with shared edges to be counted, as the added △ will become either a ◪ (with one edge added) or a ⊠ (with two edges added). In these cases, it is no longer possible to control these subgraph counts independently.

3.3.3 Adaptive resolution change mechanism

As touched upon above, GAs have long been known for the novelty of their solutions, especially in massive and unknown problem spaces [72, 94, 178]. Furthermore, the use of novelty-driven methods such as quality diversity algorithms [30] or illuminative algorithms [118], in which the aim is the creation of many diverse solutions of high fitness, often means that the focus is on the exploration of the space, even at the cost of exploitation of the

solutions found. Within these types of algorithms, the MAP-Elites method has been shown to be very effective at producing a diverse range of solutions without the need for costly archives of all of the solutions ever seen. This makes them well suited to the exploration of the massive space of possible subgraph configurations, with the directness that changes in subgraph counts can have on structural diversity making them well suited as mapping features used to define the niches [167] (hereafter called "cells", as discussed in Section 3.2). In our past work using MAP-Elites [130] we chose to use very small cells (with all cells having a resolution of 1), in order to maximise the fineness of the search and thus the detail of the possible relationships illuminated by this search. This was necessary, because without any previous work on each individual problem space, we had no prior knowledge as to the appropriate resolution to use with each space (that would yield similar numbers of realisable networks in each cell). Furthermore, with less of a need for exploitation within the cells, the optimisation of the global clustering of the network to the target level was a relatively simple problem. This meant that the exploration of the space via the generation and storage of new realisable networks was more important to our results, even at the expense of exploitation. However, setting the search to a resolution this fine meant that all mutations leading to a realisable network would be stored in the end population, resulting in an ever-increasing population as the search goes on. Although this would not have major effects on the MAP-Elites method, given that competition is only ever between individuals within a cell, it would mean the following: (1) a lower selective pressure on the generation of more diverse networks in terms of their subgraph counts; and (2) a large decrease in the optimisation pressure within each of the cells, since there are fewer solutions competing with each other (although this is not a problem here).

The challenge of choosing an appropriate cell resolution in the absence of prior knowledge about the space is one that (at the time of writing) has not been solved by researchers working with the MAP-Elites method, or indeed with any of the novelty-driven search

methods involving storage via cells [30]. Most work is in fact focused on preserving a set cell resolution for use in unbounded spaces [49] and on increasing the usefulness of these novelty-driven search methods in higher dimensional spaces [174]. This is not to say that there is no work on this problem of placing cell divisions withing the space, such as [28, 49, 174]. However these methods still focus their divisions of the space on the distribution of solutions with in the search space, in the case of [49] and [174], and do not allow the investigation of other features of interest to different research questions, with the AURORA method of [28] requiring the divisions of the space to become a "black-box" and thus hard to investigate with much detail. Similarly, the NS+LC method of [94] has been adapted to allow a adaptive reducing of the size of local neighbours compared as the search goes on and the space of solutions becomes more saturated. However again this method dose not react to the knowledge being gained during the search and thus can't be used for such detailed investigation.

Here we propose a novel solution to this problem that allows for all cells within the space to be adaptively changed in response to the of revisiting occurring during the search. This solution is defined as follows:

- each search is started at the same low resolution across all dimensions (where the number of dimensions is the cardinality of the family of subgraphs used to parameterise the networks);
- when the ratio between the number of cells being revisited (due to a mutation resulting in a solution whose associated cell has already been seen, i.e. filled, during the search so far) and the number of new cells being discovered exceeds a threshold (see Equation 3.3.3), halve the resolution (across all dimensions) of the cells with the highest measure of interestingness (MoI) (see Section 3.3.3).
- place the individual (or individuals, if more than one individual is allowed to be stored

per cell) in the new corresponding smaller cell and adjust the mutation size of mutations using this cell as a parent (see Section 3.3.3).

Three critical components of this mechanism are: (i) the MoI used to select cells whose resolution should be made higher; (ii) the choice of when to change the resolution of the cell; and (iii) the relationship between mutation size and cell resolution. We discuss and explore each of these factors below; these are also covered in Sections 4.2.1 and 4.2.2 in Chapter 4, later in this thesis.

Measure of interestingness (MoI)

In order to gain an accurate assessment of appropriate cell resolution for a problem space without prior knowledge of the space, it is necessary to gain information about the space during the search itself, ideally without the need for additional processes that could slow the search. In the case of the MAP-Elites method, mutations that lead to individuals falling within cells already known to the search (termed "revisits") are only kept if their fitness is greater than that of the current individual stored in the cell. This means that these revisits contribute only to the optimisation of the fitness of that cell. Utilising these revisits, regardless of their fitness, to inform our understanding of the space allows the resolution of our search to be adaptively changed, so that it becomes more accurate as the search goes on and evaluates more samples from the space. This still leaves the question of how best to identify areas of most interest to the search in order to best represent them in the end population. In other words, what is a significant enough difference in each of the mapped dimensions, such that any individual with at least this level of difference in one of these dimensions would be considered distinct enough from all other members of the population (and thus a "novel" solution to the set problem)? Furthermore, areas of the space with the smallest cells (i.e., the highest resolution) will have an increased likelihood of being explored, with random selection being more likely to select them based purely on their

increased prevalence in the stored population. This means that these levels will also have a large effect on the focus of the search.

Considering that this level of difference might not be uniform across the entirety of any particular mapped dimension, this is not a simple problem. The problem being addressed by the search is largely responsible for determining what should or should not be considered a "novel" solution. Therefore, a criterion is needed that reflects this diversity within a cell in such a way that it has meaningful impact on the end goal of the problem being solved. For example, here our problem focuses on those subgraph combinations leading to the widest range of diversity in network structures, avoiding areas of the space leading to non-valid solutions. As such, here we propose that this criterion should be the variance in a measure of network structure that is not uniquely determined by the subgraph decomposition of the network. As mentioned in Section 3.3.1, here we are examining only homogeneous networks and thus many structural features, such as assortativity or local clustering, that could be used for such a measure would be uninformative due to the strict nature of this degree distribution (i.e., there would be very little to no variance across valid networks for either of these structural measures). As such, here we decided to use the variance in mean betweenness centrality (BC) [52], which captures the mean number of shortest paths passing through each node in the network (i.e., a high mean BC would suggest a highly connected network with lots of nodes sharing edges). With the fixed degree of each node from the homogeneous target degree distribution, as well as the set level of global clustering, the only thing left which can affect the mean BC of a network is its more local structure (i.e., its subgraph decomposition). Nevertheless, even with a knowledge of the exact counts of the subgraphs comprising the network, there is no way to directly determine the mean BC from this information (i.e., a change in subgraph counts might not result in a proportional change in size of the mean BC). It is this feature of the mean BC that would make it an unsuitable dimension to map cells to (as shown in [167]). However, it is still a good MoI of the

networks within a cell, since it is able to highlight otherwise unnoticed structural differences. In practical terms, each cell maintains a copy of the specification of the fittest individual (since with varying resolution, the cell only specifies a range of values for each dimension of the specification), along with the variance in the MoI calculated over all individuals sampled when the cell was visited. This adds slightly to the computational cost of each mutation compared with the original MAP-Elites method, but not significantly as the variance (and all ancillary variables) can be calculated incrementally (i.e., without storing the specifications of the individuals), thereby reducing the storage cost.

When and how often should the resolution of a cell be changed?

Now that we have established how to determine the areas of the space best suited for a change in their cell resolution (hereafter referred to as a "change event"), we still have the problem of choosing when is the best time to allow these change events (i.e., at what point can we have confidence in our MoI without the need for excessive sampling of the space?). Here we propose to base this decision on the ratio of global exploration (the number of new cells discovered since the last change event, "add") to global exploitation (the number of new offspring falling within cells that have already been discovered since the last change event, "$revisits$"). Concretely, we use the following condition:

$$revisits > (Ratio \times (add + LL))$$

Where $Ratio$ and LL are constants set at the start of the evolution, which are defined in more detail below. Since we start the search with very large cells across all dimensions (in order to preserve as much of the selective pressure gained from these larger cells as possible), this approach can be thought of as evaluating the level of confidence we have in the coverage of the search space. That is, when the number of $revisits$ is greater than add, it suggests that the problem space, as defined by the current resolutions of cells across the

space, is becoming saturated and thus that we can have some confidence in our coverage of the space. Of course the ratio of $revisits$ to add is unlikely to be optimal for all problem landscapes or goals in the space and thus we allow the tailoring of this ratio using three factors: (i) the triggering ratio ($Ratio$); (ii) the lower limit (LL); and (iii) the number of cells selected for a resolution change during each change event (NC). All of these factors are described in more detail below.

$Ratio$ refers to the extent to which the ratio of $revisits$ must have exceeded add ($+ LL$, of course) in order to trigger a change event. This serves to control the overall speed at which change events occur, as well as the ratio of exploration required for a change event. A low $Ratio$ results in a significantly lower exploration of new cells being required to trigger change events and thus much more change events within the same iterations. In spaces in which it is hard to find new solutions (i.e., in which there are few solutions in the space or the solutions are sparsely distributed across the space), we would thus desire a higher $Ratio$ in order to account for this and to ensure thorough coverage of the space.

The LL is the "lower limit" and denotes the minimum number of revisits that must have occurred before a change event is allowed. For example, if $LL = 10$ and $add = 0$ then $revisits$ must be $> Ratio \times 10$ in order to trigger a change event. In this way LL is similar to $Ratio$ in that it affects the overall speed at which change events occur. However, instead of ensuring the level of exploration, it guarantees a minimum number of $revisits$ and thus samples for our MoI. If LL is too low, then change events will happen as soon as $revisits$ is greater than $Ratio \times add$. That is, if $Ratio = 1$ then this is occurs as soon as $revisits > add$, meaning that there is no guarantee of a minimum number of samples taken from the space from which to obtain an accurate assessment of the MoI.

Finally, NC is the maximum number of cells selected for splitting during each change event. For example, with $NC = 2$, two cells will be changed at each change event: the cell with the highest MoI score, and the cell with the second highest MoI score. It is worth noting that if

the MoI score of a selected cell is 0, or if the NC value exceeds that of the current population, then that cell will not be changed and the number of cells split in that change event will be lower than the NC value set. The NC value can thus be thought of as a measure of confidence in the MoI values within in the population, with a higher NC meaning that more of the high MoI cells are split more often. However, if NC is too high, it could result in the search becoming closer to random.

Figure 3.3 NC shows the effects of varying the three factors described above on a network landscape of $K = 7$ with global clustering 0.1. It is observed that with low LL and $Ratio$ values (here $LL = 0$ and $ratio = 1$), a greater number of cells are created; conversely, a low LL results in a far fewer cells than a low $Ratio$. These additional cells are at the same level as those seen with the low LL, suggesting that here a low $Ratio$ does not promote "drilling down" (e.g. lots of change events in the same region of the space) on particular cells, but rather a greater spread of cell divisions across the space. This is to be expected in this landscape in which the MoI (here the variation in BC; see Section 3.3.3) is very closely related to the features being mapped (here the genome, i.e., the population count of chosen subgraphs; see 3.3.1). This means that the level of variations should naturally decrease with the size of the cell, making it much more likely that larger cells show the highest MoI values. This, combined with the low expected variation in BC as a result of the tightly controlled fixed degree distribution (homogeneous here) and global clustering, means that without thorough sampling of the cells (via revisiting) there is little likelihood of drilling down on any one point. This therefore favours even divisions across the space, such that no one cell will have the number of revisits required to show more than the expected level of variation associated with that particular size of cell.

Further comparing the number of solutions found in either case in the same number of iterations, although more solutions are found with the low $Ratio$ (consistent with the increased number of smaller cells in this case), the number of solutions is not proportional to

the number of extra cells created. In fact, there is a difference of only 184 solutions despite the extra 80 cells split in the low *Ratio* condition, which equates to a difference of almost twice the number of solutions per cells split (with a average of 4.9 solutions per change event for low *Ratio* compared to 8.6 for solutions per change event for the low LL, as shown in Figure 3.3).

This further suggests that in the low *Ratio* condition, the choice of which cells to split (i.e., their location in the feature space in which the resolution is increased) is significantly less informed than in the low LL condition. This means that the cells split seem to have less relation with the density of valid solutions in the space.

Similarly, when we look at cases with high LL or high *Ratio*, there is the expected lower number of cells (with the high Ratio cases suffering more), with a greater focus on areas of higher variation. However, there is still no drilling down on any particular cells, with the cell size still going no lower than 16. This means that although there is now a lot of revisiting, the number of larger cells is still high enough that the expected variance from their size is enough to obscure any areas of greater interest. This conclusion is confirmed when we examine these high LL and *Ratio* conditions after the same number of change events (i.e., 58 for LL and 138 for *Ratio*). In these cases, we see a minimal cell size of 8 is reached by both conditions, but only after a much larger coverage of the space compared with the mid value condition. When we investigate varying the levels of NC, we see that with low NC, there is of course a much lower number of cells created and that the positioning of these cells tends towards areas of higher solution density. Furthermore, as in the lower LL and *Ratio* conditions, there does not seem to be any drilling down on particular cells. However, when we examined the same condition after more solutions were found, we did see splitting to a minimum cell size of 8, with 500 solutions found. This is despite the fact that the cells were still all focused around the same areas of higher solution density.

Comparing both high and low conditions to the min value condition ($LL = 3$, $Ratio = 2$,

and $NC = 2$; centre plot in Figure 3.3), we see that when all these values are controlled within the correct range for the landscapes, we start to observe a drilling down to a minimum cell size of 8, without requiring a more complete coverage of the space (i.e., only 448 solutions are found). These results are of course limited to this one set of network constraints, but we believe that for the homogeneous networks explored here, the problem landscape is unlikely to differ to the extent that these values of LL, $Ratio$, and NC affect the level of diversity seen in the results. There could be some justification for an increased NC, or a decreased LL and/or $Ratio$ to account for the larger search spaces that result from larger values of K. However, these larger spaces are better dealt with by allowing an increased number of change events to happen on their own, rather than increasing the rate at which changes occur based on these values, given that in all cases, the confidence in the MoI and the likelihood of $revisits/adds$ is unchanged/unknown.

Relationship between mutation size and cell resolution

Finally, using the adaptive method of cell resolution change described above, the search space (or at least how it is represented in the end population of solutions) is radically changed by focusing on the areas of most interest. These areas of interest are unlikely to be evenly distributed across the space (as seen in Figure 3.3) and thus a disparity in the size of cells across the space is quite likely. As discussed above, these changes will have an effect on the balance of exploration vs exploitation in the end population stored for these differently sized cells. As the search goes on with decreasing cell resolution across the space, we would want to ensure that this is reflected in the way that the search moves around the space. Here we suggest linking the range of mutation size to the resolution of the parent cell, specifically to 1 to 2 times the cell resolution of the individual. For example, a cell with resolution 64 could only have mutations within the range 64 to 128 applied to create its offspring. This results in a method of adaptive mutation that is heavily linked to the mapped space

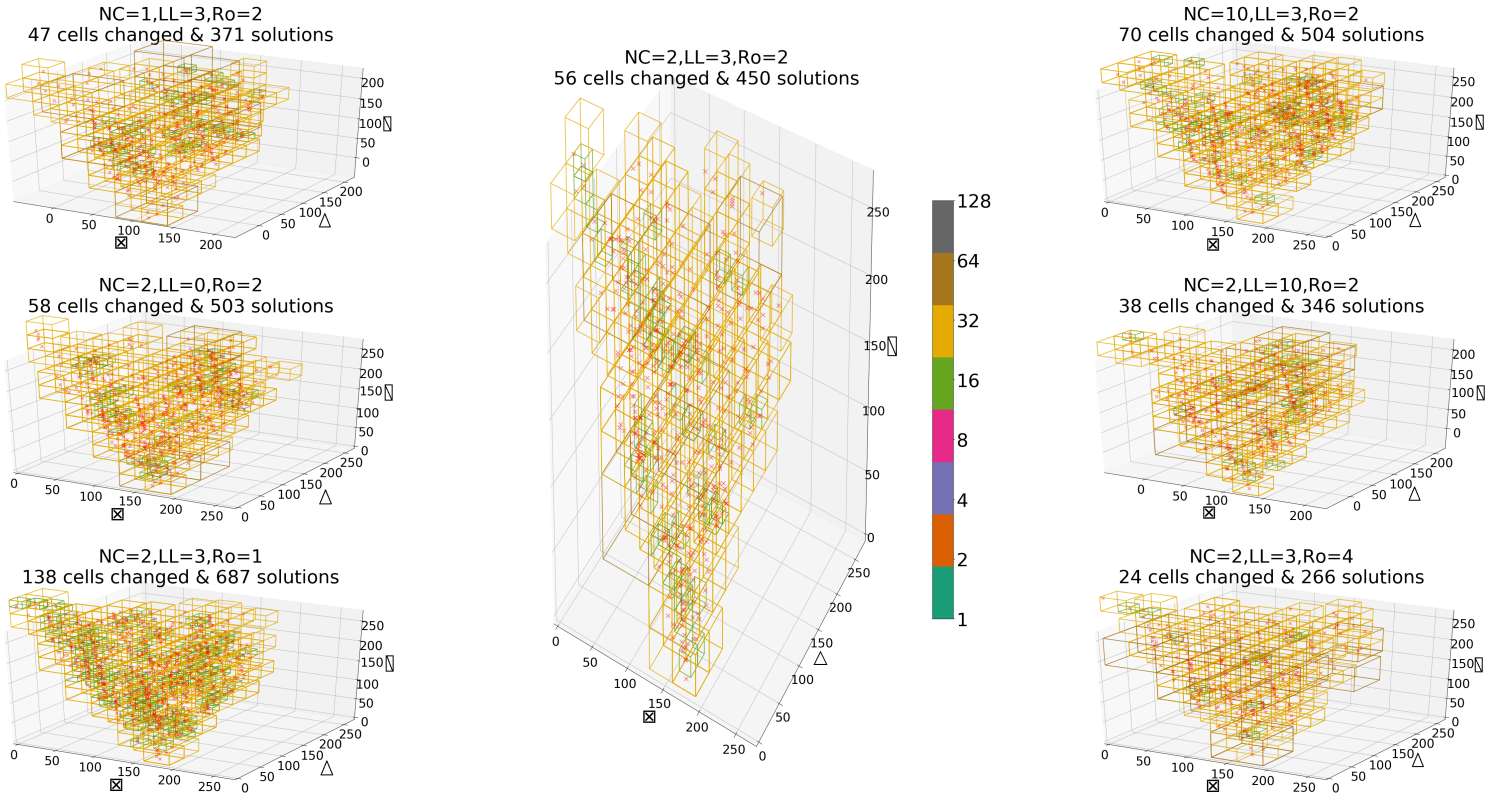


Figure 3.3: Space of solutions found (red dots) for all clustering-inducing subgraphs (△, ⊠, and ◪) after 2,000 mutations for networks of degree $K = 7$ and global clustering of 0.1. The size of each of the cells holding these solutions is shown in a colour corresponding to their size (see bar in middle plot), starting at a size of 64. The values of NC, LL, and $Ratio$ are given above each plot for the seven conditions.

being searched and, with the proposed method of adaptive cell resolution given above, also the level of MoI throughout the space. Furthermore, this method is independent of any prior knowledge of the space, meaning that it is equally applicable to any space or distribution of the MoI that might be found in it. For example, in a space with an evenly distributed MoI, in which there is no particular focus on any of the cells and thus the average cell resolution drops steadily, the average mutation size will decrease steadily with progressive mutations, similar to the way temperature decreases in simulated annealing. However, if the MoI is unevenly distributed (i.e., with some cells ending up with significantly smaller resolution than others), the average mutation might not decrease greatly, but rather only sharply in the areas with the highest MoI values. In all cases, this method allows dynamic control of the balance of exploration vs exploitation as the search goes on, with decreasing mutation size increasing the level of exploitation within the areas of highest MoI (or increasing exploitation more generally in the case of a more even MoI distribution) as the number of mutations increase. The value of this kind of adaptive mutation mechanism in controlling the trade-off between exploration and exploitation is well established [76], including within the MAP-Elites framework [126]. The decision to keep the range of these mutations such that they focus exclusively outside of the parent cells (i.e., a minimum mutation of $1\times$ the resolution of the selected cell means that any offspring will always fall outside of its parent cell) was made in order to capitalise on the lack of any optimisation within the cells for this problem space. Here, unlike in the majority of MAP-Elites methods, revisiting is only used in the measurement of the MoI (i.e., there is no optimisation of the individuals within the cell), meaning that we keep the first example we find from each cell. Thus, by focusing our mutations outside of the parent cell, we can maximise the time spent exploring new cells. This improves the overall coverage of the space, whilst accounting for the lower rate of revisiting via the increased LL/lower $Ratio$ (and/or lower NC), without detriment to the search.

Implementation

Being an iterative method of optimisation, GAs are vulnerable to computationally expensive methods of evaluation resulting in very long run times. This is especially a problem when the focus on achieving a good coverage of the space requires many more iterations over the space, such as is the case with MAP-Elites [53]. Here, we use the CMA method of subgraph allocation to realise networks from which the features of a genome are evaluated (see Section 3.3.1). CMA is a combinatorial method and therefore requires a number of attempts in order to find a suitable arrangement of subgraphs. This means that in the worst case scenario, CMA could require 100 repeats of its allocation process before a suitable arrangement is found, if any. In addition, we repeat the full CMA method five times with the same genome in order to attempt to average any small differences in network created. The CMA method was not developed with computational efficiency in mind and was in fact adapted from the MATLAB programming language to Python for this research (Python being the sole language used for this work). Thus, the number of generations we could run in any of the conditions shown here was limited by the available computer resources. This problem could be addressed in future work by using some kind of modified surrogate fitness, such as those used in [53, 53], which would reduce the number of times the CMA needs to be run. However, this was not the focus of this work and would still require testing ground truths about the feature space that were not fully known at the start of this work (i.e., that changes in subgraph counts will predictably change evaluated features across the space). Moving on, although the idea of starting with a coarse discretisation and then increasing the granularity was mentioned by the authors of the MAP-Elites framework [118], we are not aware of any existing implementation of this at the time of writing (note that the SHINE method of [157] was published around the same time as the first submission of this paper; see Section 4.2 of Chapter 4) . Indeed, even in those papers in which cell size was a point of

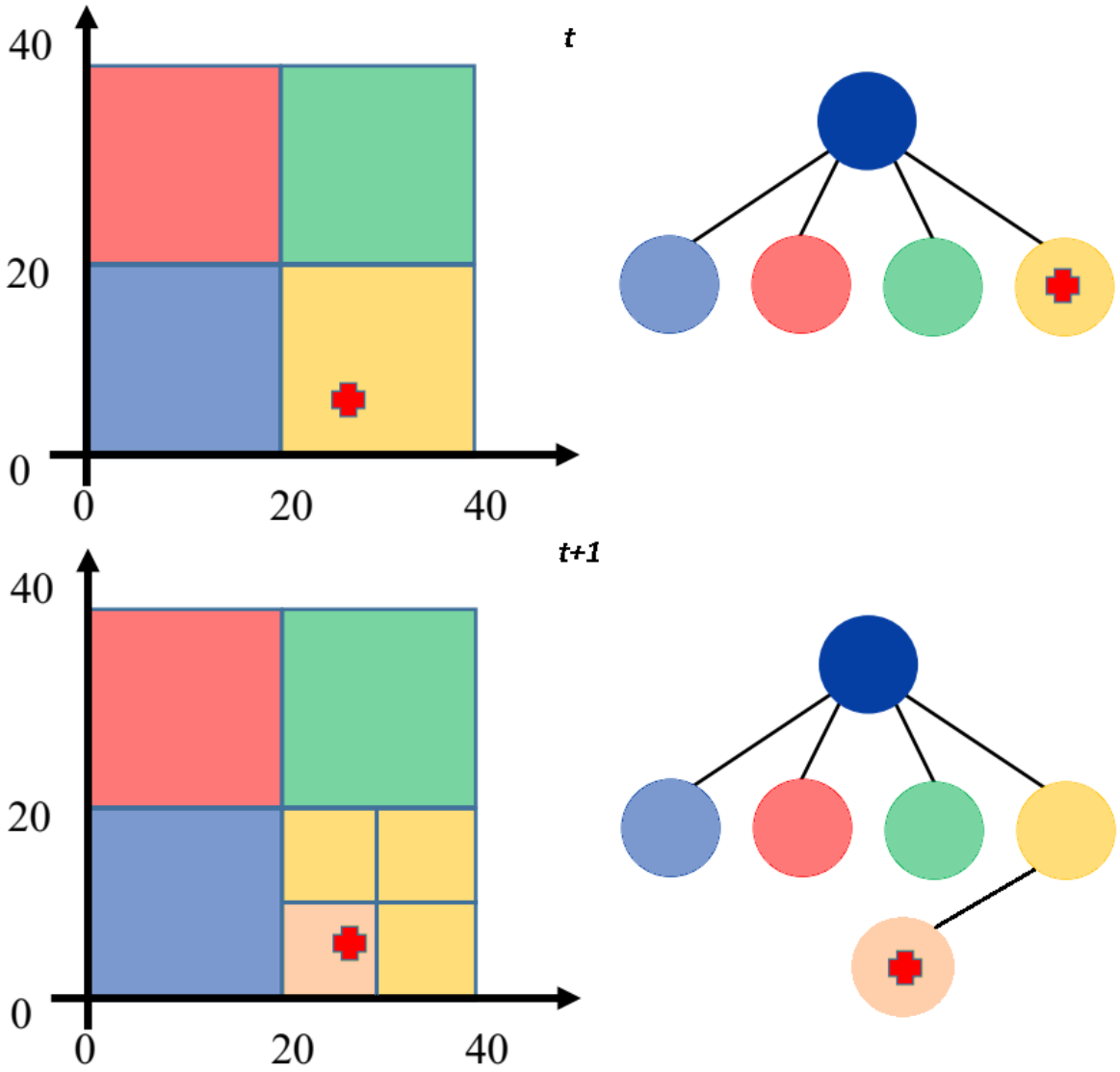


Figure 3.4: Cell map structure for a two-dimensional space (right side) and its accompanying tree data structure (left side), before and after a cell resolution "change event" (t before, and $t+1$ after). The red cross shows the location of the stored individual for that cell in the space. It should be further noted that this method only stored solutions for cells that have been filled, in order to save on storage and computational costs.

interest (e.g., [177, 176]), the total number of cells was set at the start of the search via prior knowledge of the space, often with a emphasis on having as few cells as possible (see Section 3.3.3), and remained unchanged throughout the search. For a mechanism such as ours to be computationally practical in a high-dimensional search space, there is a need for efficient operations for searching the population as well as adding and updating cells. Our implementation relies on a tree data structure (see Figure 3.4 developed in-house and available at https://github.com/harrygcollins/TreeBasedGA; see [25]). This allows not only for efficient searching of cells in the population but also for cells in the population at one level of resolution to be easily split to accommodate new cells in the same feature space location at a higher resolution (i.e., the splitting of one cell into smaller cells within the same space). This can occur without the need for labelling the cell via an additional linked list. This has a significant effect on the efficiency of the implementation, particularly in the case of this method as, unlike MAP-Elites, we do not have a fixed size for the stored population, so cannot rely on allocating a fixed size of memory or a fixed indexing order.

3.4 Results

Here we start by examining the effect our method has on a range of target network problems in order to show its effectiveness in informing our understanding of the space of possible network structures (see Section 3.4.1), before moving on to explore in more detail the impact of our methodological changes in terms of three measures:

- rate of discovery: the number of iterations needed to obtain a particular number of networks (see Section 3.4.2);
- quality of discovery: the diversity in network specifications uncovered (see Section 3.4.3);

- behavioural diversity: the significance of the quality of discovery on behaviours with real-world dynamics (see Section 3.4.4).

Unless stated otherwise, all runs of the GAs shown below were started with cells of size $64 \times 64 \times 64 \times 64 \times 64$, the five subgraph repetitions in encoding as the mapped features, and the search variables set to $LL = 3$, $NC = 1$, and $Ratio = 1$ to determine change events.

3.4.1 Examination of scaling values of target clustering and edge density on the difficulty of valid network exploration

In order to establish the effectiveness of our method at improving our understanding of the space of valid network structures, here we explore the kinds of networks found after a relatively short search (i.e., the first 4,000 networks found) for a range of target networks with different global clustering and/or degree.In this paper we do not examine the effect of changing the size of the network (i.e., the number of nodes involved) as, although this is a question of some interest in the field, because of the computational limitations of CMA discussed above in Section 3.3.3. We avoid this issue by keeping the network size to the same size of 1,000 nodes in all of the examples described below. We thus focus our investigation on varying the levels of edge density within the networks generated by setting the number of edges based on our choice of degree distribution (for a regular network with $K = 5$, 7, and 9) and the density of triangles via target clustering (0.1 and 0.2). As stated in Section 3.3.1, because of the need to prevent the genome being forced to allow subgraphs that do not contribute to a fixed number of edges in the encoding (i.e. overlapping of edges), we can only go as far as clustering 0.1 with homogeneous networks of $K = 9$. To clarify, to obtain a clustering of >0.2 with a regular network of K9 degree, there must be at least 2,400 triangles present in the network. In order to then gain a high number of triangles, the encoding of the subgraph-included in the network would have to be able to account for subgraphs that are created via the use of a shared edge (i.e., an edge involved in another

encoded subgraph). Furthermore, it would be possible to investigate clustering values above 0.2 with regular networks of K5 or K7, but this would involve increasing the computational strain on the CMA method. This would result from there being far fewer arrangements that can accommodate the subgraphs encoded in a graphable network of the given degree distribution, and as such, this possibility is not explored further here in this paper.

All conditions were run with the same starting setup as stated above, with the starting population created for each of the five conditions (see the Appendix for a description of the starting populations). When we examine the number of networks found vs the number of iterations taken to find them (i.e., the rate of discovery, accounting for iterations that led to revisiting a cell already seen or to a genome that was outside the set range of each subgraph; see Section 3.3.1), we begin to see a marked difference from the expected 1:1 rate (green "control" line in Figure 3.5) if all of the networks encoded were able to produce a graphical network via the CMA method. Furthermore, these differences are significantly different for each of the conditions. This deviation from the control is due to "un-realisable" genomes, meaning that in 100 attempts of hyper-degree sequences, CMA was unable to find a connected network that kept the same degree distribution and/or global clustering (see Section 3.3.1) for the proposed genome made during that iteration. Even with our more exact movement through the space via Diophantine mutations, it is to be expected that there will still be some areas of the subgraph space that cannot be realised via the CMA method (or even by any method). This is due to the limitation of the target degree distribution and, to a lesser extent, of the target global clustering (though this would be more likely with much higher clustering than the 0.2 shown here) . Thus the differences seen in Figure 3.5 could be thought of as a crude measure of the "difficulty" of the space of realisable networks, at least in terms of finding realisable networks. In terms of the effect the target clustering has on this difficulty, it would seem to be less of a factor than the degree. However, the overall effect does seem to be dependent on the distribution, with the K7 conditions showing very little

differences with clustering 0.1 or 0.2 compared with the K5 conditions, which seem to show a small but significant difference between clustering 0.1 and 0.2. This would be consistent with the higher clustering requiring more ⊠ and ◪ subgraphs to be fit to increase clustering, requiring more nodes with three edges to fit these subgraphs. Specifically, in the $K = 7$ condition this is less of an issue as each node is able to fit two of these subgraphs, whereas in $K = 5$, there is only room for one such subgraph per node. This suggests that the difficulty decreases with increasing K size, as can be seen with the K7 networks (blue and light blue), which are both very close to 1:1, compared to the K5 networks (red and magenta), which are significantly slower. Of course the exception to this is K9, which is dramatically slower than both the K7 and K5 conditions. This is likely due to the limitation discussed above, namely the need for the subgraphs to not be forced to use overlapping edges to fit within the degree of the network. This is consistent with the much larger variation in iterations required for K9 compared with the other two K value conditions (i.e., K5 and K7).

Examining how the search landscape is divided up over the three clustering-inducing subgraphs (△, ⊠, and ◪) in Figures 3.6, 3.7, and 3.8, we can see that K9 shows very large cells over the whole of the space. This suggests that, despite having the highest number of iterations to get to 4,000 networks, there have been far fewer revisits in the K9 condition than in the other conditions. This is clearly because the larger value of K implies a greater space of possible subgraph configurations, as we can also observe in the other four conditions. This would seem to lend support to the idea stated above that the difficulty of the space increases with a decreasing K value, and that in the case of K9 there must be a different factor affecting the difficulty, which is whether or not the networks are realisable with the CMA method. Further to this, gaps across several parts of the space in the K9 condition (notably between 448 to 512 and 832 to 896 in the △ space, as well as in a few parts of the ◪ space; Figure 3.6) would suggest that there is some factor(s) affecting the ability to find solutions in those areas of the space, as in all the other conditions there is good coverage

across the space of all three subgraphs. This could be an effect of the K9 space of solutions being much larger than that of the others and thus less well represented in the 4,000 networks seen here. However, we would still expect to see greater coverage of these gaps in the $\triangle$ space, given the emphasis of our method on undiscovered regions of the space. Additionally, when we look at the K7 conditions (Figure 3.7) we see that, even when comparing the lowest number of iterations gained over the three different K conditions (K5, K7, and K9), the K7 conditions have the greatest number of revisitings of the solutions added. This is shown by the fact that they have the largest number of cell divisions, with three instances of cell size 1 in K7 clustering 0.2 (no such instances are seen in any of the other conditions shown here, with the K5 conditions going down to cell size 4 at the lowest, and K9 to cell size 8). This is counter to the observed size of the solution space in each of the K conditions, with the larger spaces of K7 compared to K5 meaning that, if all solutions are equally easy to find, there would be greater likelihood of revisiting in the smaller K5 solution spaces. Again this suggests that the difficulty identified in Figure 3.5 correctly reflects some true features of the different conditions as they are searched by our method. Examining the effect of clustering at this level also confirms some of the statements made based on Figure 3.5; for example, that the effect seems highly dependent on the degree of the target network. This is demonstrated by comparing the ranges of the subgraphs between the two K5 conditions (Figure 3.8), which increase significantly in the higher clustering condition (thus increasing the size of the space and the difficulty of the search). In contrast, the two K7 conditions there is a much smaller difference in the ranges.

Based on these results, we chose to focus all of the following examinations of the impact of our changes to the methodology on K7 regular networks. This allows for greater confidence that the results we obtained are due to the direct effects of our method, as opposed to difficulties within the search space.

3.4.2 Impact of Diophantine-based mutations on the rate of discovery

To characterise the impact of the use of our Diophantine-based mutations (herafter referred to as "exact mutations") on the search process, we compared it with a baseline of random mutations. Setting the cell resolution to its maximum (i.e., one cell per network specification, $cellsize = 1$), we systematically varied the mutation size from 2–128 in powers of 2, in order to fit with the mutation sizes used for linking mutation size to parent cell resolution. For the random mutations, a random number was selected from a range of $\pm$ the maximum count for each of the subgraphs (as specified in Section 3.3.1), which was then added to a randomly selected individual from the population. In all cases, we evaluated the rate at which new (valid) networks were discovered as a function of mutations (4,700 in all cases) as well as the diversity in network specifications (the coverage). As shown in Figure 3.9, there is a significant gain in speed and in the number of networks obtained when exact mutations were used (whereas random mutations found only 18 networks), irrespective of the mutation size. It is worth noting that a higher rate of discovery does not necessarily result in a greater number of networks. This is because a substantial number of iterations are lost, either due to out-of-bounds mutations or a higher rate of revisits.

Looking at the differences between each of these sizes of exact mutations, we can see that there is a clear need to balance between the larger mutation sizes (128, 64, and 32) and those that are too small, as seen with size 2. With the larger mutations the changes imposed cause a proportional change in all non-mutated subgraphs in order to accommodate them (as described in Section 3.3.2). This therefore causes a loss of locality (i.e., excessively large mutations lose the benefit of a locally heterogeneous resolution), thereby causing their search to become closer to a random search. On the other hand, with excessively small mutation sizes, the problem becomes that of being able to move outside of the areas already seen (i.e., excessive revisiting). This is shown in the case of mutation size 2, which, despite

having the slowest rate of new cell discovery, was still able to find far more valid networks in the same number of iterations compared with mutation sizes 128, 64, or 32, suggesting that its low rate of discovery is due to revisiting (given that boundary errors are unlikely with such a small mutation-induced change). If we look closer at the mutation size with the highest rate of discovery (size 8), we see that the same pattern is true, with sizes 16 and 4 being within a standard deviation of each other in terms of their rate of discovery. Likewise, the same pattern is repeated with size 16 being faster, but finding fewer valid networks, compared to size 4. The drastic drop in rate seen in 2 is likely a result of how few exact mutations are possible to account for such a small change (i.e., if we add two △, then the only way to accommodate that change is to add one □ and to remove one ⊠ and one ⬠). Figure 3.10 shows the frequencies at which subgraphs occur for both exact and random mutations. Whilst random mutations show fairly uniform frequencies, coverage of the space is extremely patchy due to the difficulty of obtaining realisable networks (as discussed in Section 3.4.1). In contrast, exact mutations lead to dense coverage of the space (including beyond that sampled by the random mutations). This figure clearly shows the impact of mutation size, with small mutations (e.g., size 2) resulting in well-defined peaks of higher frequency, while large mutations (e.g., size 128) yield a more uniform histogram (although the number of networks found drops significantly with increasing mutation size, as explained before). Therefore, there is a balance to be reached via the adaptive resolution mechanism, which is explored in more detail below. In the following analyses, all experiments start with a cell size of 64, because a cell resolution of 128 was found to lead to too few networks (i.e., it seems to be a far too coarse a resolution to be suitable for the fixed-resolution method used for comparison).

3.4.3 Impact of adaptive resolution search on quality of discovery

To illustrate the benefit of using an adaptive rather than a fixed-resolution search, we compared the network specifications discovered by our method with those obtained using either a fixed mutation size of 64 (the coarsest resolution possible that enabled the greatest coverage) or a fixed mutation size of 8 (which was shown previously to yield the highest rate of discovery). As shown in Figure 3.11, networks uncovered using the adaptive resolution search show the largest breadth of subgraph counts (e.g., the largest range of ⊠). Interestingly, even though using a fixed mutation size of 8 yields a much larger ensemble of networks (almost 20× larger than using either our method or a fixed-resolution size of 64), the distributions of subgraph counts are fairly unimodal, reflecting the lack of coverage. Furthermore, with the adaptive resolution search the distribution of counts are kept very similar to those found in the fixed mutations for all of the clustering-inducing subgraphs (△, ⊿, and ⊠). The difference in the distributions of non-clustering inducing subgraphs (□ and ⬠) suggests that our choice of MoI had the possible effect of pushing the search to focus on increasing the range of clustering-inducing subgraphs over non-clustering-inducing subgraphs.

To address the concern that such differences may be a random artefact, we assessed the range of BC found in the above networks and that of an identical number of networks generated using BigV and dk-2.1 randomisation. Both methods were used in order to maintain the same distribution of global clustering coefficients (see right panel in Figure 3.12). This was achieved by stopping the BigV rewiring process when the required global clustering coefficient was reached, and by seeding the dk2.1 randomisation using CMA-generated networks with the required clustering coefficient. Nevertheless, as shown in Figure 3.12 (left panel), we found that the generated networks span different ranges of BC values. The most likely explanation for this effect is that the CMA sought to prevent subgraphs around a node

from sharing edges (see [151, 150]). Nevertheless, this result demonstrates that the networks are structurally different. To our knowledge, this is also the first evidence that, despite the claims made by its authors, dk2.1 randomisation may not provide uniform sampling.

3.4.4 Effect of the quality of discovery on behaviours of real-world dynamics: complex contagion and Kuramoto models simulation

To illustrate that the diversity found in these networks (i.e., the differences in their higher-order structures) does impact behaviour, we consider two measures of network-based dynamics: the complex contagion model [111] and the Kuramoto model of synchronisation [89, 40] .

The complex contagion (CC) model, which is a modified version of the susceptible-infected-recovered (SIR) infection model [111, 128], is intended to describe the spread of more complex behavioural patterns, such as rumours or the use of technology, (i.e., behaviours for which the "infection" of a node requires more than one exposure to the infection). In an SIR epidemic, a susceptible node connected to an infected node becomes infected at rate τ, and once infected it recovers at rate γ, independently of the network. All processes are considered to be independent Poisson processes. The final epidemic size is computed by counting the number of recovered individuals at the end of the epidemic when no further infected nodes remain. Each individual epidemic starts with one single infected node. This model differs from a classical SIR epidemic by requiring that susceptible nodes are exposed to multiple infectious events before becoming infected. Furthermore, these events must be from different infectious neighbours as only the first infection attempt from an infectious node counts, and infected individuals remain infected for the duration of the epidemic. These dynamics are known to exhibit a critical transition in relation to the number of infected nodes at the start of the epidemic. Preliminary work in the lab showed that given a degree distribution and a global clustering coefficient, the parameter value at which the

transition occurred could fluctuate [111].

Here, we compared the range of parameter values at which the transition occurred for maximally different (defined by the Euclidean distance between their subgraph counts) pairs of networks: (*a*) using random exploration; and (*b*) using our proposed search mechanism. For each of the networks and for each parameter value, we ran 100 simulations to robustly identify the critical transition. Figure 3.13 shows a substantially greater range of parameter values for networks found through the proposed search mechanism, thus confirming that greater diversity was achieved with our method (even though the computational cost of eliciting the same number of networks through random mutation was far greater).

The Kuramoto model is a classical model of synchronisation [1, 22] and has been used to study the oscillatory behaviour of neuronal firing [81, 16] among many other biological systems. The Kuramoto model describes the phase behavior of a system of mutually coupled oscillators with a set of differential equations. Each of N oscillators in the system rotates at its own natural frequency ωi, i $= 1, \ldots, N$, drawn from some distribution g(ω). However, it is attracted out of this cycle through coupling K, which is globally applied to the system. Time t is taken to run for T seconds of length $dt = 0.01$. The differential equation to describe the phase of an oscillator is: [88]

$\Phi_i(t) = \omega_i(t) + \frac{K}{N}\Sigma_{j=1}^{N} sin(\Phi_j(t) - \Phi_i(t))$

As for the results shown above for the CC model, we compared the maximally different pairs of networks found for the *a* and *b* methods when run on the Kuramoto model (see Figure 3.14). This shows a significantly larger difference between the networks gained from our proposed search mechanism compared with those gained from the random search method, again confirming the diversity shown in the CC model results.

3.5 Discussion

In this paper, we presented a methodology for exploring structural diversity in networks sharing a set of properties. Encoding networks based on their subgraph decomposition provides control over the local structure around nodes. The experiments described here reveal that our methodology makes it easier to elicit structural differences between these networks which have an impact on dynamics running on the networks. In contrast to classical network-generating methods, which rely on very long mixing times to provide uniform coverage, our approach borrows concepts from evolutionary algorithms to more rapidly identify interesting regions of the solution space, namely regions of the space containing structurally more diverse networks.

Our results demonstrate the need for more knowledge regarding the spaces of possible network structures that only share set clustering and degree distributions, given that the current methods, which were previously thought to sample the space without bias, appear not to sample from the full range of network structures (see Figure 3.12). One promising line of enquiry for this could be to systematically study the importance of a given subgraph on dynamics by restricting the search process to mutations that increase/decrease the number of instances of this subgraph. This could then be paired with more detailed analyses of the cell resolution distribution over each of these subgraphs, in order to establish the importance of different subgraph regions to one another and to the MoI chosen. Although we only provided examples in which the degree distribution and global clustering coefficient were specified, the framework described here is applicable to other scenarios. In fact, with the inclusion of optimisation during the search, there is no reason that something like a fixed level of assortativity or other structural features could not be imposed on the population. However, this would require more focus to achieve this fixed level within a cell, before adding to the MoI and deciding the cells to be selected for splitting; that is, the MoI should

not be based on non-valid networks to avoid invalidating this measure. This would also open up the possibility of using other structural measurements for the MoI, such as local clustering diversity or vertex-level entropy, which would significantly affect the kind of networks focused on during the search (see Figure 3.11). Of course, this would require further sensitivity analysis, because while the analysis performed here (see Section 3.3.3) does provide guidance for future researchers to implement their own networks with tailored parameters, the results shown are limited to the networks examined here. Thus, there is the possibility of performing further testing on a wider range of target networks and/or MoIs in order to maximise performance. One future direction of this work might even be to use yet more computationally expensive measures, such as the results of behavioural dynamics, to constrain the networks generated via the use of surrogate fitness methods of illumination (see [53, 60]). This kind of surrogate modelling could also be used as a way of increasing the size of the networks being explored. This might mean treating each of the cells as its own pool of surrogate fitness/MoI for the individuals inside, decreasing the resolution of the cells as the accuracy of the fitness/MoI is established. However, how this might affect the sensitivity of the factors controlling "change events" is difficult to predict without experimentation.

3.6 Appendix

Starting populations for all of the conditions run, with the genome given in the order [△, ⊠, □, ⬠, ◪] and K representing the number of degrees in the homogeneous network (i.e., K5 means all nodes in that network have 5 edges connecting them):

K5, clustering 0.1 = [[35, 54, 134, 166, 41], [29, 66, 138, 173, 20], [65, 56, 131, 167, 22], [17, 37, 128, 159, 84], [45, 4, 114, 141, 136]]

K5, clustering 0.2 = [[36, 65, 58, 69, 185], [26, 146, 89, 110, 28], [64, 136, 83, 103, 29], [300, 64, 44, 53, 55], [472, 34, 25, 27, 29]]

K7, clustering 0.1 = [[62, 65, 126, 155, 189], [22, 166, 167, 207, 7],

[296, 58, 111, 138, 86], [362, 47, 103, 129, 75], [64, 67, 124, 158, 184]

K7, clustering 0.2 = [[76, 259, 27, 38, 144], [92, 307, 48, 58, 40], [84, 280, 37, 46, 98], [74, 255, 27, 35, 153], [110, 214, 9, 13, 217]]

K9, clustering 0.1 = [[38, 39, 73, 89, 503], [44, 65, 82, 102, 448], [704, 87, 59, 72, 74], [92, 184, 125, 158, 186], [908, 51, 35, 42, 44]]

Parameters of evolutionary algorithms

number of mutations per iteration = 1

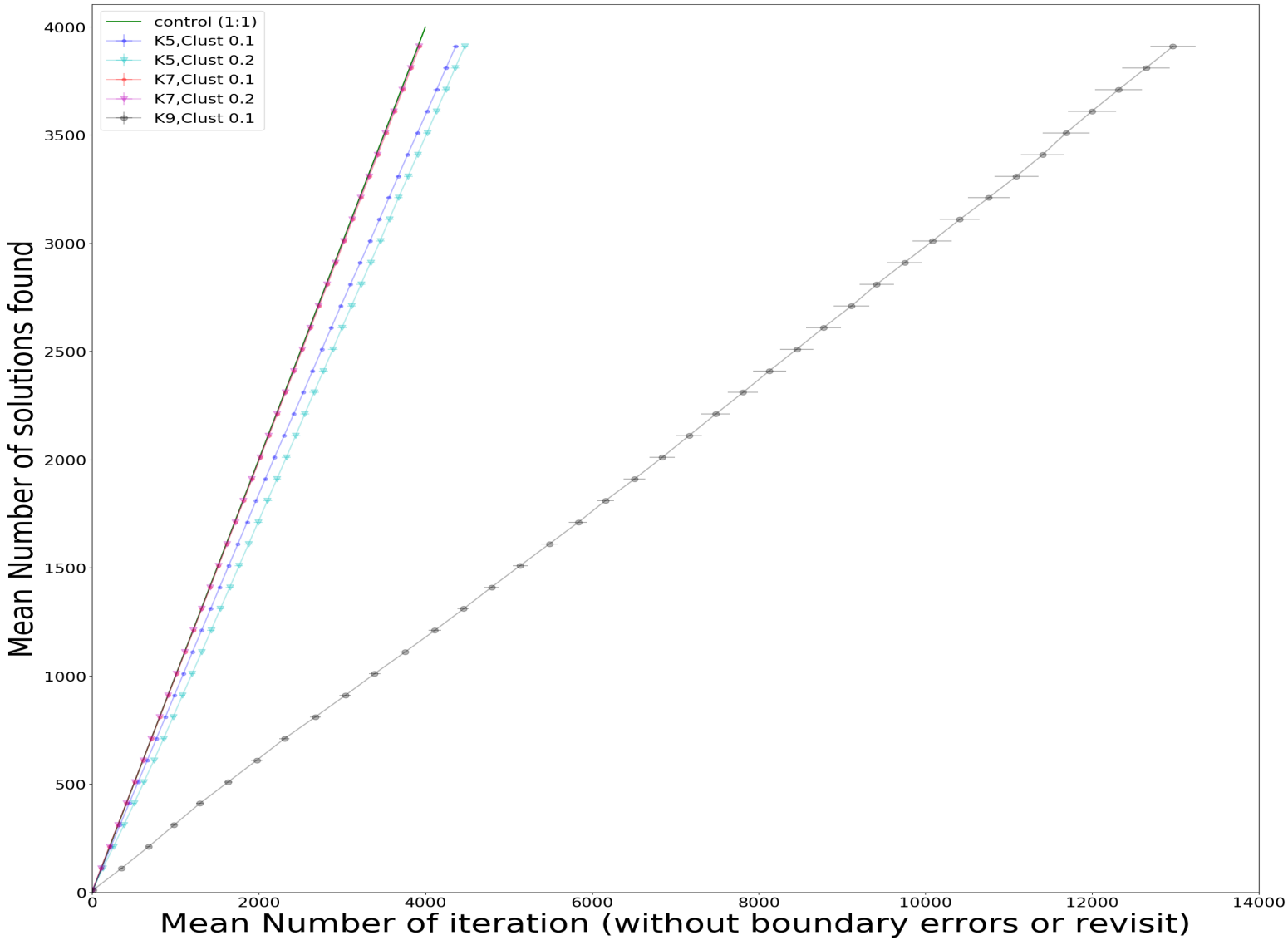


Figure 3.5: Mean number of solutions found vs mean number of iterations without revisits or boundary errors (i.e. without iterations that led to revisiting a cell already seen or a genome that was outside the set range of each subgraph; see Section 3.3.1) for five conditions with varying K and clustering values ($K = 5$, clustering 0.1; $K = 5$, clustering 0.2; $K = 7$, clustering 0.1; $K = 7$, clustering 0.2; and $K = 9$, clustering 0.1) after 4,000 networks were found. Each condition was repeated five times with the same starting population for each condition.

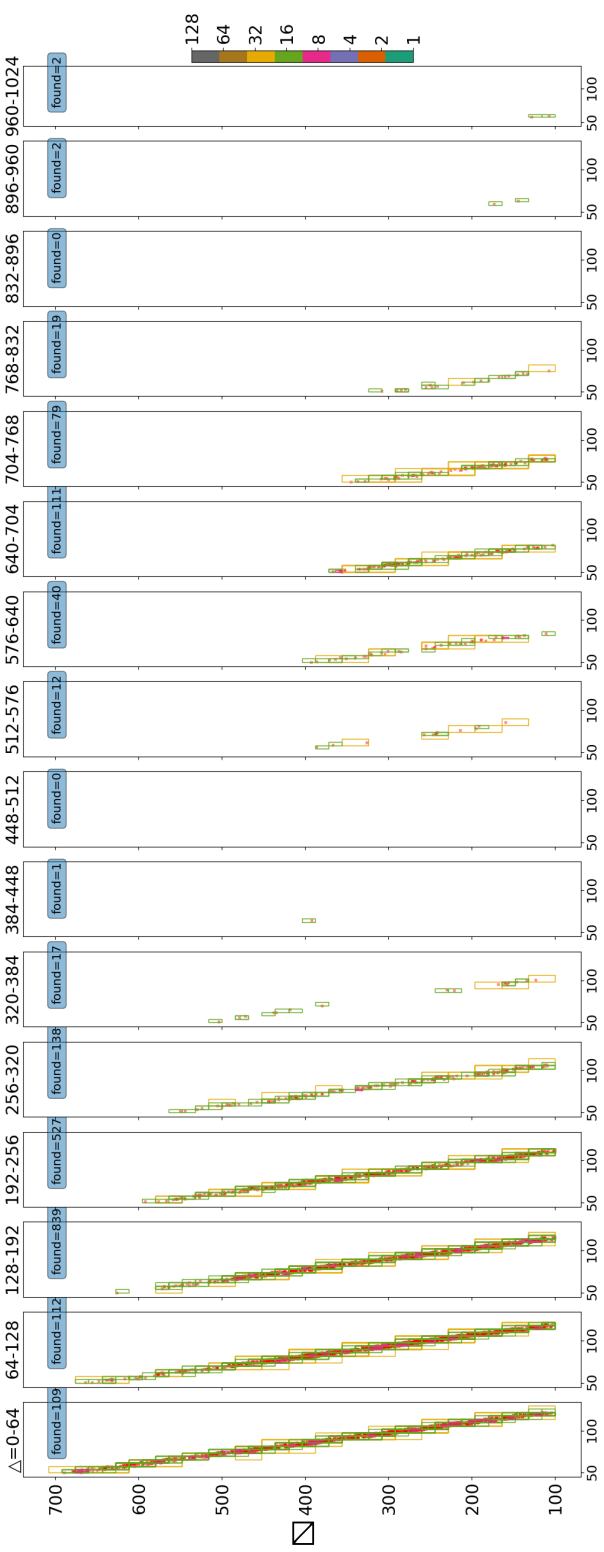


Figure 3.6: Positioning of valid networks (degree of K9 and clustering 0.1) found (red crosses), along with their cell sizes (colour-coded according to size using the colour scheme in the right-hand bar), after 4,000 iterations across the three clustering-inducing subgraphs. Each panel shows the networks found within the given range of △ values (shown at the top of each panel), with ⊠ shown on the X axis and □ on the Y axis.

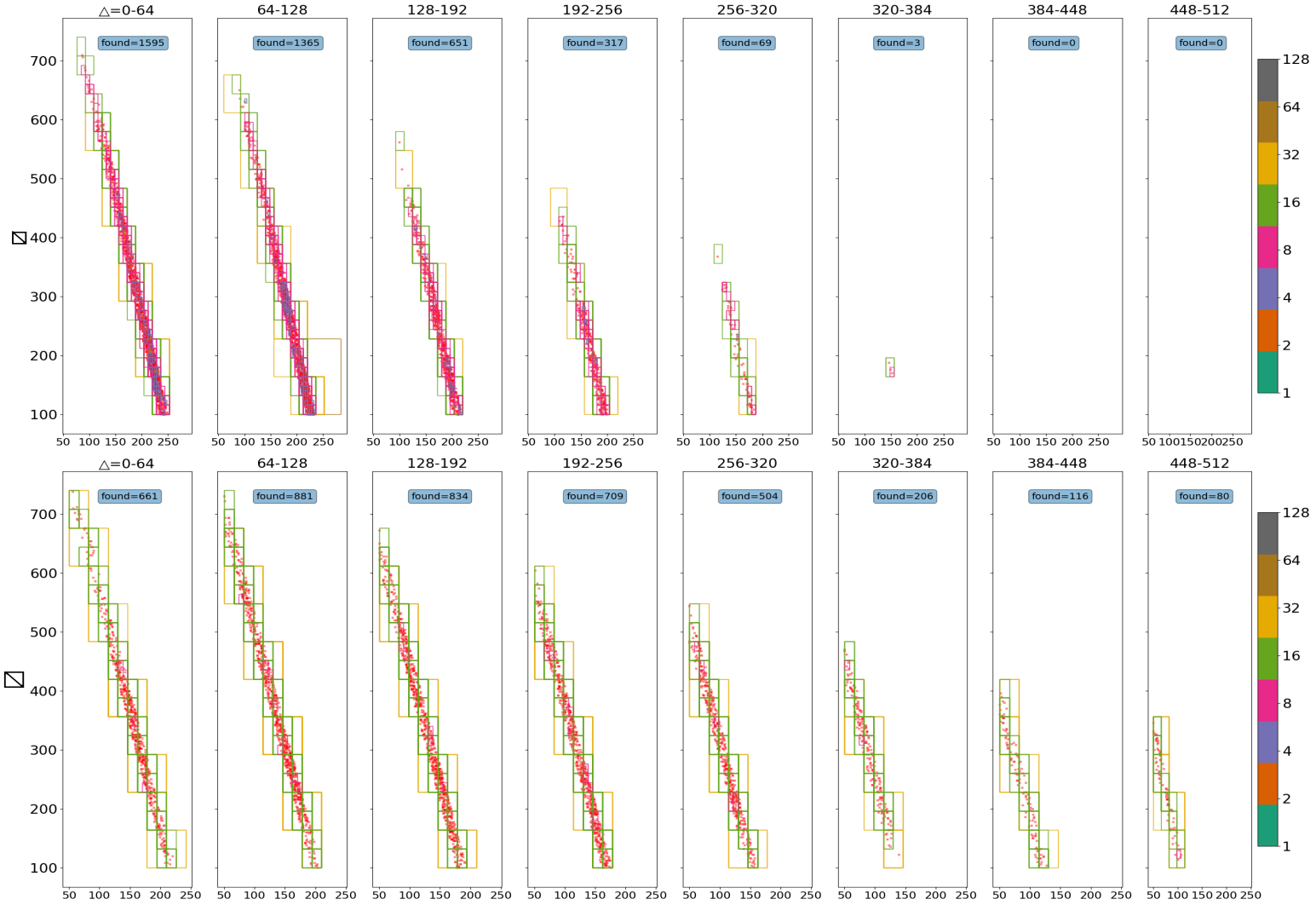


Figure 3.7: Positioning of valid networks (degree of K7 and clustering 0.2 (top) or clustering 0.1 (bottom)) found (red crosses), along with their cell sizes (colour-coded according to size using the colour scheme in the right-hand bar), after searching 4,000 networks across the three clustering-inducing subgraphs. Each panel shows the networks found within the given range of △ values (shown at the top of each panel), with ⊠ shown on the X axis and ⧄ on the Y axis.

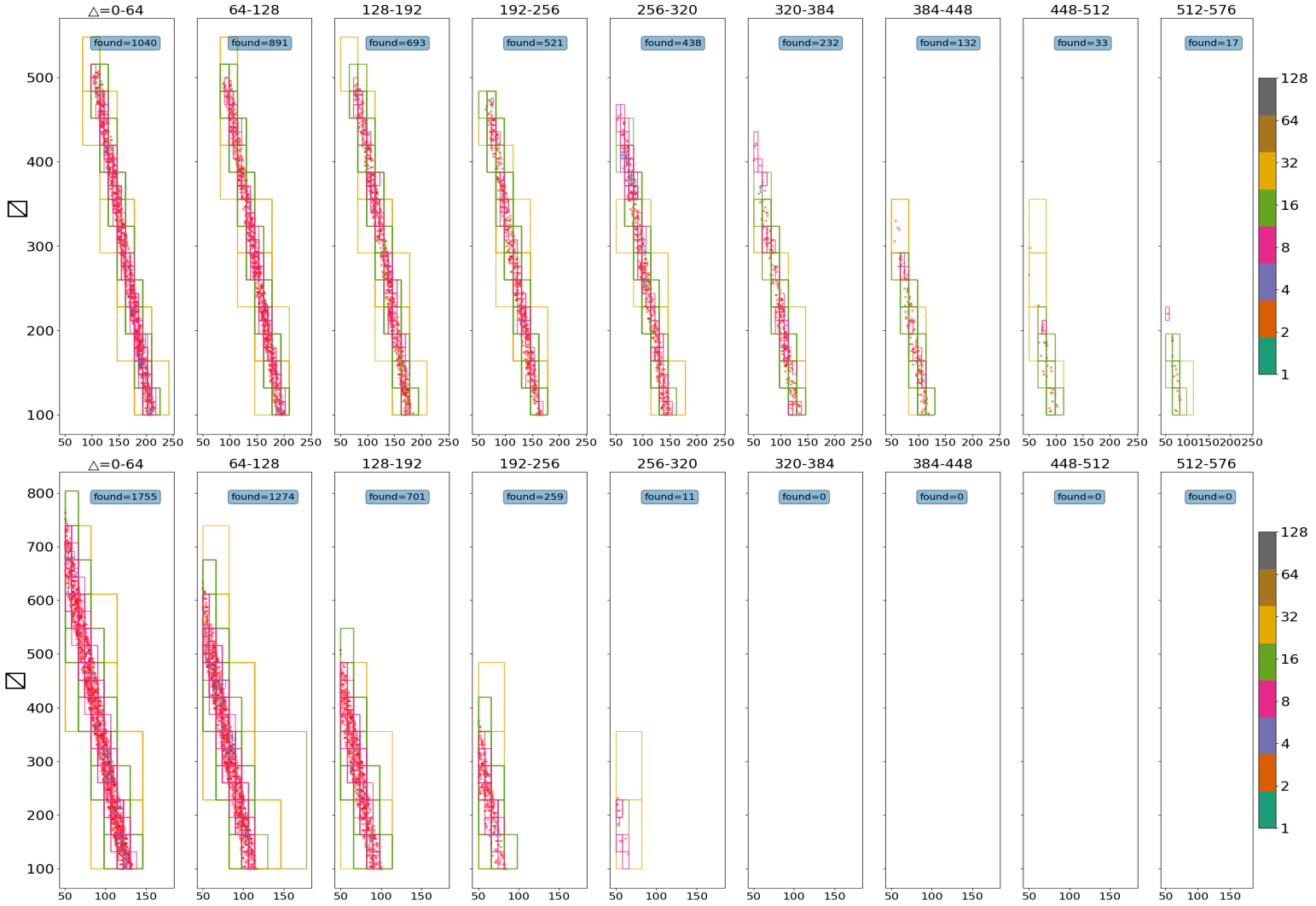


Figure 3.8: Positioning of valid networks (degree of K5 and clustering 0.2 (top) or clustering 0.1 (bottom)) found (red crosses), along with their cell sizes (colour-coded according to size using the colour scheme in the right-hand bar), after searching 4,000 networks across the three clustering-inducing subgraphs. Each panel shows the networks found within the given range of △ values (shown at the top of each panel), with ⊠ shown on the X axis and ◪ on the Y axis.

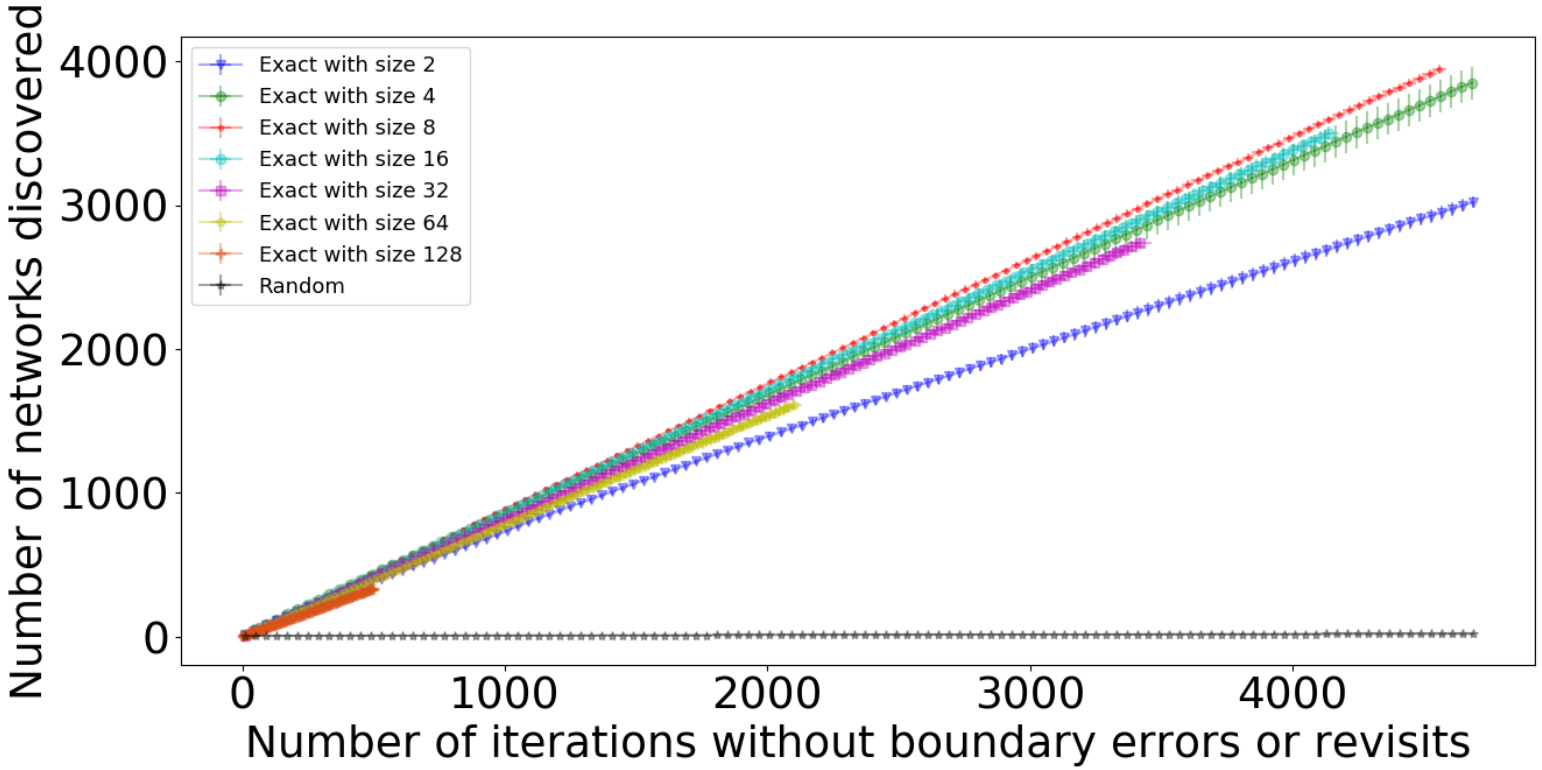


Figure 3.9: Rate of discovery of valid networks when exact mutations (with different mutation sizes) and random mutations are used. All conditions are shown after 4,700 iterations, with all iterations ending with a revisit or with boundary errors being removed.

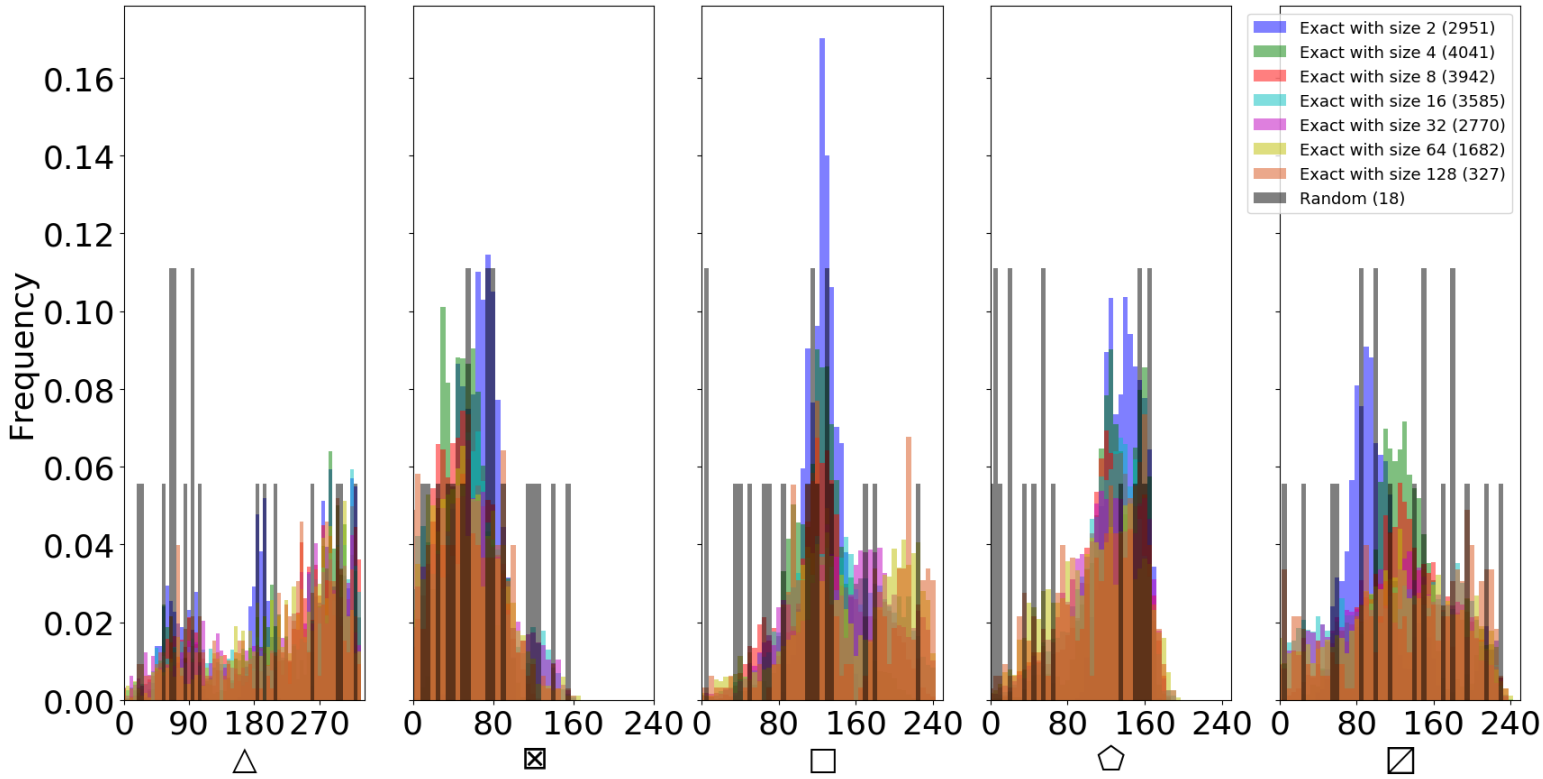


Figure 3.10: Frequencies of subgraph population counts for both exact mutations (with mutation sizes ranging from 2 to 128) and random mutations. Bin size was set to 5 for all subgraphs. The number of networks found by each method after 4,700 iterations is provided in the legend.

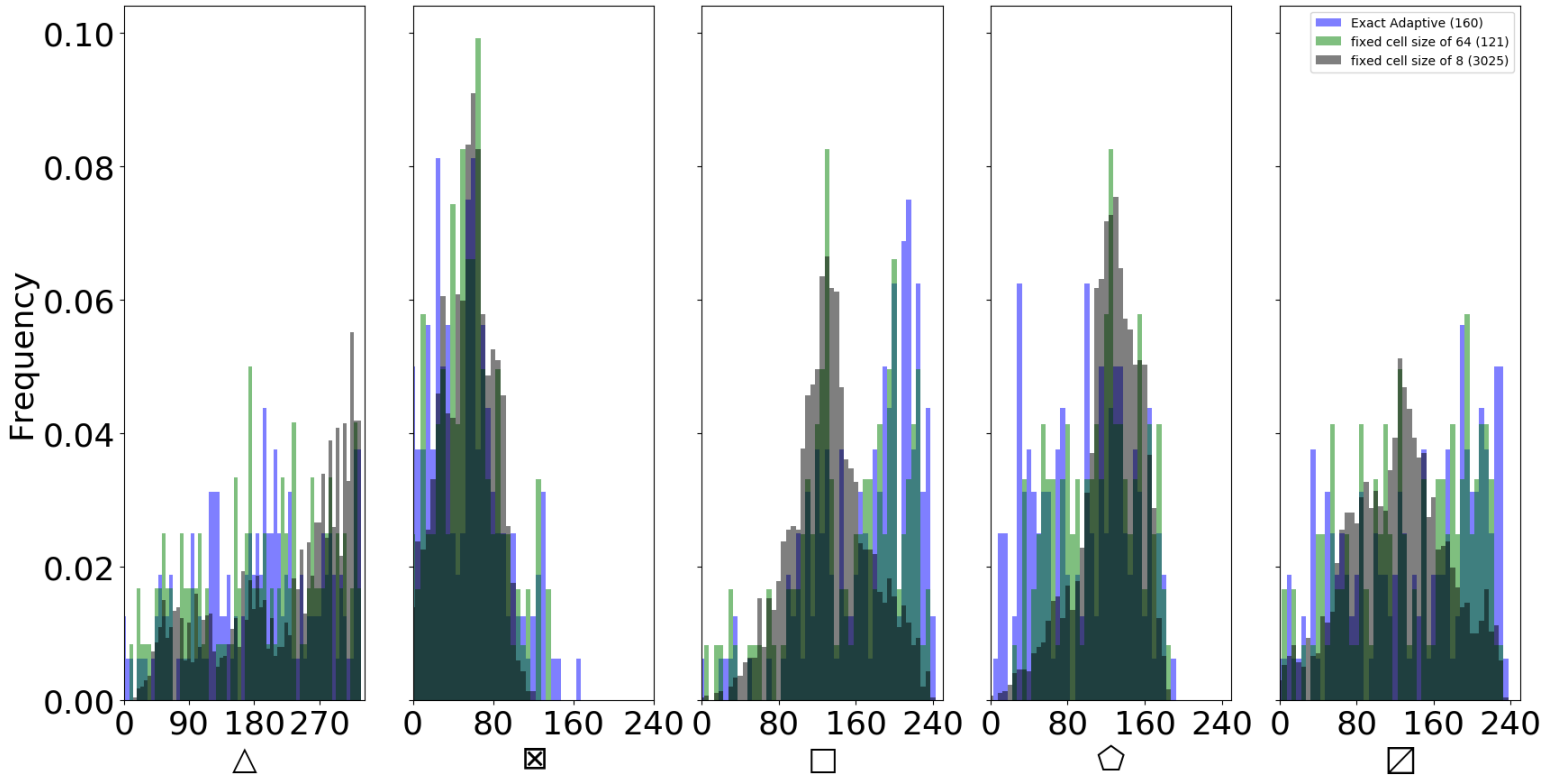


Figure 3.11: Frequencies of subgraph population counts for the proposed method (blue), or using a fixed mutation size of 64 (green) or 8 (black). Bin size was set to 5 for all subgraphs. The number of networks found by each method after 4,700 iterations is provided in the legend.

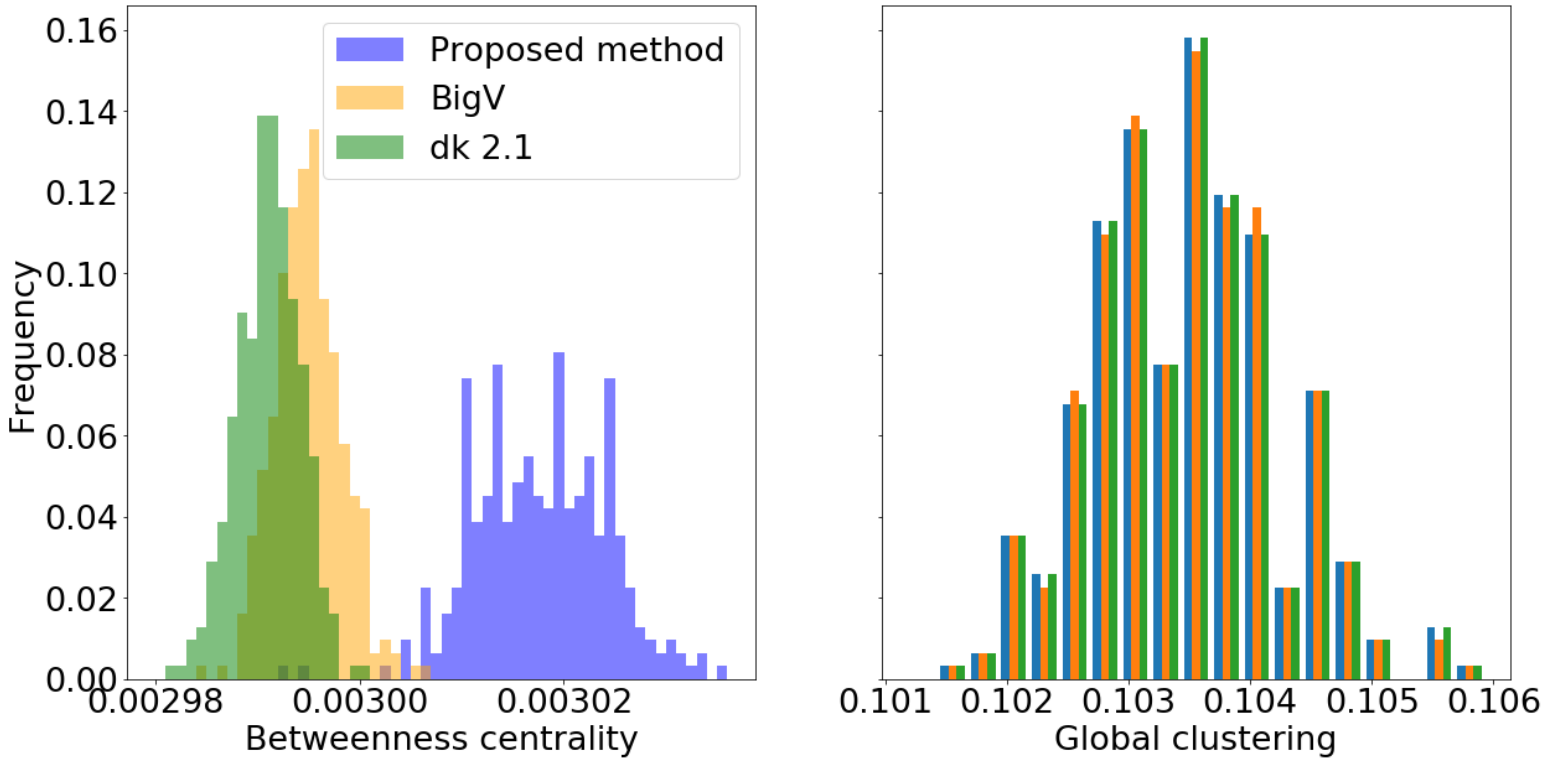


Figure 3.12: Histograms of betweenness centrality (left) and clustering coefficients (right) for GA, BigV, and dK-generated 5-regular networks.

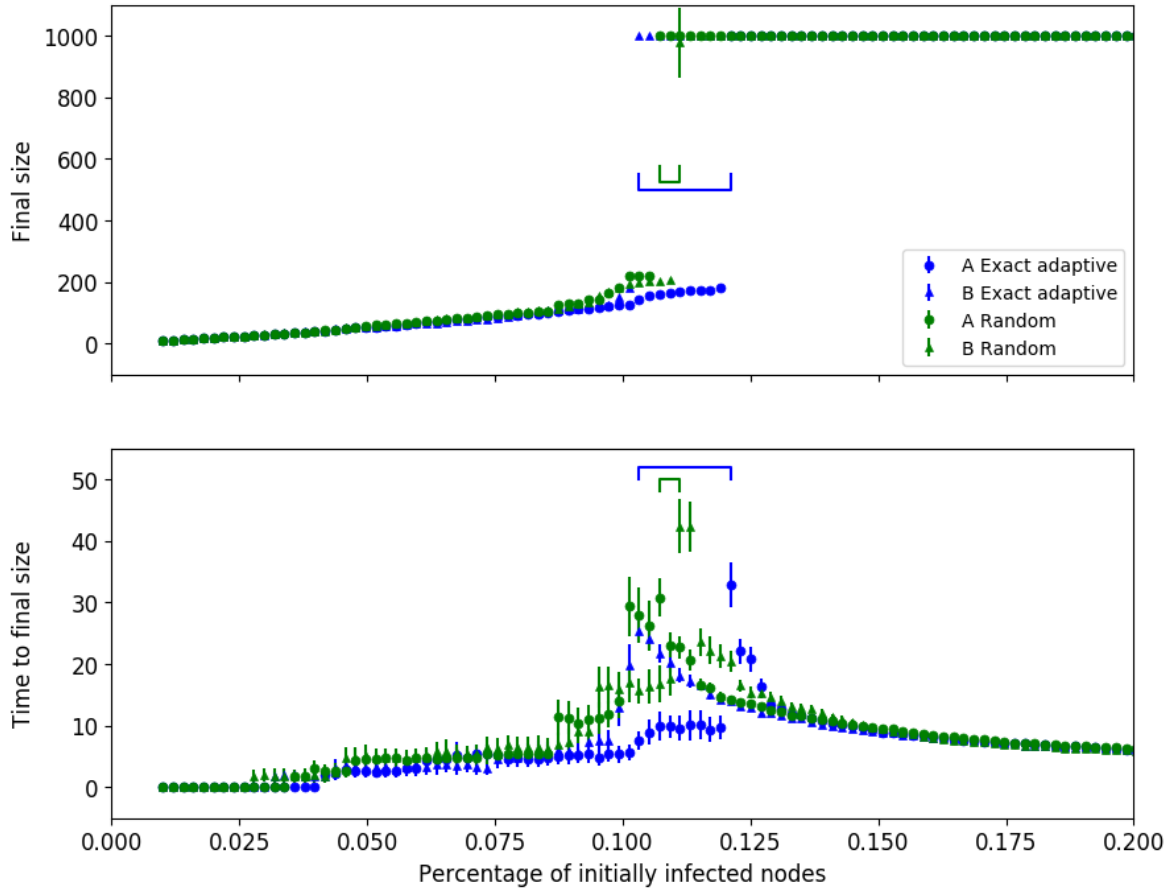


Figure 3.13: Ranges of parameter values over which the critical transition of complex contagion simulations occurs in networks found by the proposed search mechanism (blue) or by random mutations (green). The critical transition is identified as the value parameter at which there is maximal variability in both the final size and the time needed to reach this final size. All $N = 1000$ individuals had a threshold $r = 3$ and a transmission rate $\beta = 1.0$.

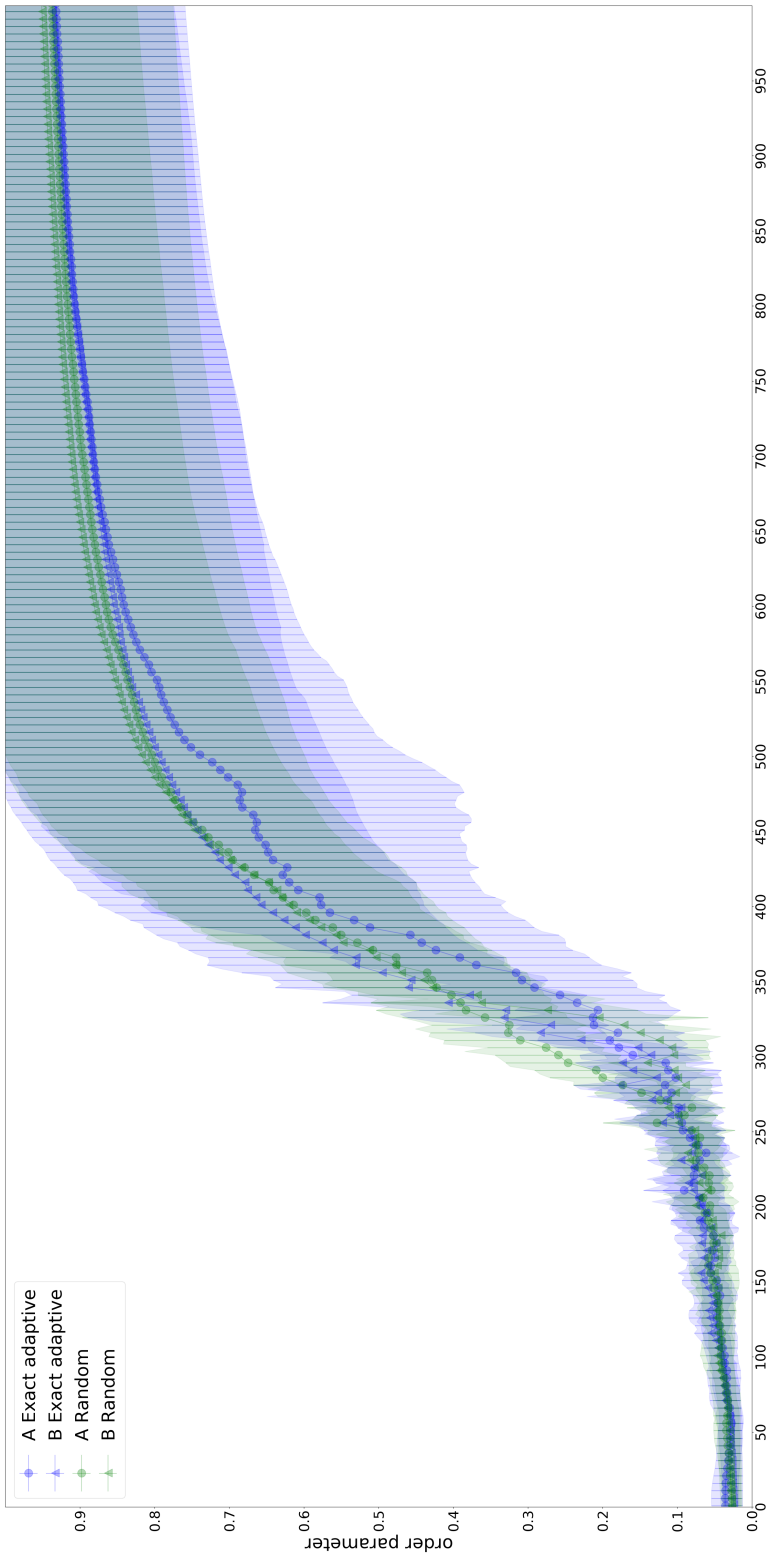


Figure 3.14: Evolution of the global order parameter with increasing coupling constant for networks found by the proposed search mechanism (blue) or by random mutations (green).

Chapter 4

Illumination of unknown feature spaces via adaptive resolution change MAP-Elites (ARC MAP-Elites): A general framework

[1]

[1]This chapter is intended to be submitted for publication as soon as possible, but is currently unpublished at the time of thesis submission.

4.1 Introduction

In many fields there are problems for which multiple solutions of equally high performance exist across distinct regions of a search space. For such spaces, exploration of the full range of distinct global optima can be of great relevance not only in the investigation of the relationship of the search space to solutions (such as in the investigation of large feature spaces [130] or complex fitness landscapes [158, 172, 106]), but also in providing a behaviourally/phenotypically diverse population.This often extends the versatility of these solutions, enabling them to be applicable to many situations [31, 28]. This need for sampling from multiple areas of the space is not just important for identifying the regions of these global optima, but also for helping to prevent the misleading effects of deceptive landscapes, in which local optima can result in sub-optimal solutions and/or dead-ends in the optimisation path [136]. Recently the importance of focusing on a wider exploration of the space, even at the cost of exploitation, has been investigated with methods such as novelty search [92] showing that even a wholly exploration-driven search can be effective in the optimisation of certain problem landscapes. Pairing this search for novelty with a local level of optimisation (i.e., by limiting competition to the nearest neighbours) allows for greater identification of areas of high fitness, but also an increased coverage of the solution space, without the need for excessive evaluations of the entire space of solutions [92, 93, 118]. In early work on novelty search with local competition (NS+LC) [93], this combination of a more localised level of optimisation with a reward for solutions that are unlike the stored population was achieved by limiting competition between individuals to a preset value K of nearest neighbours in the search or behaviour space. The average distance from the K nearest neighbours of an individual was also used as a measure of the novelty of the individual in the population. Thus, the fitness of each individual compared to that of their K nearest neighbours was used as an additional optimisation aim. Specifically, the

average distance between neighbours and the number of neighbours whose fitness is lower were maximised, and making this a multi-objective optimisation.

This combined fitness creates a niching effect that maximises fitness, whilst at the same time encouraging individuals in the population to spread out across the space of possible behaviours and/or genomes. Indeed, as individuals take advantage of lowered competition in the less seen regions of the space of solutions, they gain a selective advantage based on the novelty they bring to the population. This way of encouraging population diversity has been shown to be effective [171, 96, 139, 138], even when focusing solely on genome differences. This drive to increase the diversity of solutions as each individual pushes to increase their distance from one another does have the desired effect of increasing diversity of the solutions, but has the issue that, since the method has no view of the space as a whole (i.e., a global view), it can end up "cycling" back and forth from one area of high diversity to another as the population changes (see [118, 30]). This issue becomes more likely to occur the longer the search goes on, leading to some areas of the space being unexplored and thus making good coverage of the space less likely.

The desire to ensure a greater coverage of the space of solutions inspired the development of illumination algorithms, notably including the Multi-dimensional Archive of Phenotypic Elites (MAP-Elites) algorithm [118]. In this algorithm, rather than treating the problem as a multi-objective optimisation, novelty is ensured by dividing the space of solutions into a grid with a set number of cells along the feature dimensions in which novelty is desired (termed the "mapped dimensions").

These cells are treated as "diverse" groups for the purposes of local competition and storage of the populations, such that only the individual with the highest fitness in each cell found will be kept in the end population [118]. This addresses the problem at a much more global level and thus prevents the "cycling" seen in NS+LC, as well as providing a faster measurement of diversity [30]. Despite this, both of these methods rely heavily on having

some prior knowledge of the space being searched. This is required in order to determine the diversity of different solutions, with K and the size of cells across the space both being determined via some relatively arbitrary judgement of the space prior to the exploration of the landscape. In the case of MAP-Elites, this means that even though the difference between two solutions in terms of distance in the space could be very small, they could both be treated as diverse solutions within the stored population. Furthermore, both methods assume that solutions in the space are uniformly distributed across the space, with possible solutions at every point of the space, and that diversity across the space is of equal interest throughout the space. For example, if the distribution of possible solutions tended more towards a normal distribution, then the degree of interest in the far ends of the space would be higher compared to that of those more abundant solutions in the centre of the distribution. This is because those solutions in the centre would represent a much larger percentage of the stored solutions and thus be more likely candidates for selection with the current method. Even when the assumption of uniform distribution is true for a landscape, both methods still lack the ability to react to growing knowledge about the diversity of the space gained during the search. NS+LC tends to "cycle" with an increasing number of generations, while MAP-Elites revisits the same cell solely to optimise the fitness within that cell. In other words, if a revisit does not result in a higher fitness than that of the individual already stored, then that revisit adds nothing to the search as a whole.

To mitigate these limitations, we propose an extension of the MAP-Elites method to allow for adaptive resolution changes (ARCs). Specifically, in our proposed method, the size of niches (cells) across the search space is reduced through "change events" that refocus the population on the areas of greatest interest to the search problem, determined by a measure of interestingness (MoI), as our knowledge of the space grows with re-sampling/revisiting the niches (cells).

In the following sections, we describe the proposed framework , analysing all of its

component factors and illustrating the sensitivity of the method to these factors in a number of known 2D landscapes (for simplicity of visualisation) of varying ruggedness. We then benchmark our method against the original fixed cell size MAP-Elites method by applying it to two problem landscapes: (1) a hexapod robot walking problem, which is a commonly used example of the use of MAP-Elites [29, 78, 132, 173, 41, 177]; and (2) an exploration of the space of networks sharing fixed structural features, which was previously explored via this method [131], as well as in Chapter 3 of this thesis.

4.2 Methods

MAP-Elites, as previously mentioned, is a grid-based genetic algorithm (GA) that focuses on encouraging diversity within the population by dividing the solution space into predefined regions, referred to as "cells". Competition is then limited to occur only locally within each of these cells. This control over the solutions stored encourages diversity by ensuring that whenever a new region of the space (i.e., empty cell) is discovered during the search, the new solution is guaranteed to be stored. This solution can then only be replaced if a subsequent solution is also found within that cell (henceforth referred to as "revisiting") and has a higher fitness.

It is also important to note that the encouragement of diversity in the stored solution can be in terms of the search space (the genome or genomic space) and/or the behaviour space (phenotype space). The space searched is determined by the choice of the dimensions used for the division of the space of solutions, which are henceforth referred to as the "mapping dimensions".

The standard MAP-Elites algorithm, first introduced in [118], has a very straightforward implementation of this process (see Algorithm 1). First, the mapping dimensions are discretised into N cells that each represent a "novel" solution or group of solutions in the feature space or "niche". This has the advantage of allowing an efficient measure of the

novelty of all solutions without having to compare them to all of the other solutions (or even the k nearest neighbours, as in the NS+LS method).

Algorithm 1 Pseudocode for the default MAP-Elites algorithm, taken from [118]

1: ($P \leftarrow 0$, $X \leftarrow 0$) ▷ Create an empty, N-dimensional map of elites: (solutions X and their performances P)
2: **for** iter = 1 → I **do** ▷ Repeat for I iterations
3: **if** int < G **then** ▷ Initialize by generating G random solutions
4: x'← random_selection()
5: **else** ▷ All subsequent solutions are generated from elites in the map
6: x ← random_selection(X) ▷ Randomly select an elite x from the map X
7: x'← random_selection(x) ▷ Create a randomly modified copy of x
8: b'← feature_description(x') ▷ Simulate solution x' and record its descriptor b'
9: p'← performance(x') ▷ Record the performance p' of x'
10: **if** P(b') = 0 or P(b')<p' **then** ▷ If cell empty or occupant's performance is ≤p', then
11: P(b') ←p' ▷ Store the performance of x' according to descriptor b'
12: X(b') ←x' ▷ Store the solution x' according to descriptor b'
13: **return** feature-performance map (P and X)

During each iteration of the algorithm (i.e., the selection and mutation of a member of the population; see line 5 onward in Algorithm 1), each of these cells is restricted to storing only one solution with the highest observed fitness so far, although others have relaxed this condition [141]. This means that each cell acts as a niche population, and thus exploitation of the population can be thought of as only occurring inside of the cells (i.e., mutations that lead to new individuals that fall into cells already seen). In contrast, the exploration of the population consists of finding new cells that have not been seen before. This makes the size of the cells an important feature of the balance between exploration and exploitation, with the size and positioning of cell divisions determining the maximum diversity that can be elicited in the end populations. Simultaneously, both the size and position of these cell divisions across each of the dimensions used to map them also have a large effect on the level of selective pressure within each of these niches and within the population as a whole. For example, too many cells in a part of the space (i.e., many small cells) will lower the chances of competition in that part of the space, thus weakening the selective pressure,

which might result in a population with a lower fitness. As such, the main concern of the field thus far has been to keep cells as large as sensibly possible. For example, the centroidal Voronoi tessellations (CVT) method [174] employs such tessellations to set a fixed number of cells regardless of the number of dimensions in the descriptor space and avoids increasing the number of cells, which in turn would reduce their size.

As stated before, however, this still means that the size and positioning of the cells is based entirely on prior knowledge of the space being searched, with all areas of the space being treated identically.

Thus, in an unknown space there is a need for some way of measuring the level of meaningful diversity that can be gained across each of the mapped dimensions; that is, the amount of change in any one of the mapped dimensions resulting in a meaningful difference to some measure relevant to the investigation of the space. Assessing this MoI across the mapped space and then applying it to the cell sizes such that each cell size is changed based on additional information about the space (i.e., from further sampling during the search) is a non-trivial problem.

There has been some work using a hierarchical spatial partitioning of the mapping space, based on the sparsity of solutions in the space (notably the SHINE method [157]). However this work focuses only on encouraging coverage of the rarest solutions and assumes that there is some difference in the density of solutions around the dimensions being mapped. This assumption rules out the use of such methods when the genome space is used as the mapped dimension.

Here, we propose preserving the size of cells as much as possible by starting with a handful of very large cells that are reduced only in the areas that have shown the highest interestingness, as measured by a predetermined MoI. The size of possible mutations from such a cell is also linked to the size of each cell, which results in the level of the search being scaled with the cell size. This means that we are able to preserve the size of the cells as far

as possible, decreasing it only in the areas of most interest to our search, and more accurately representing any varying level of diversity across the space of possible solutions. Furthermore, this means that we rely much less on prior knowledge of the space being searched compared with the fixed size cells of MAP-Elites. Of course, prior knowledge of the space is not an all-or-nothing matter and any knowledge can be useful in tailoring the search for greater efficiency. As such, we include three critical components that allow this shaping of the resolution change based on any additional prior knowledge of the space or problem being searched, namely: (1) the MoI used to select cells whose resolution we wish to focus on; (2) the choice of when to change the cell resolution; and (3) the relationship between mutation size and cell resolution. Below we detail each of these factors and give examples of their effectiveness in a range of 2D fixed problem landscapes, demonstrating their effect on cell divisions across the space.

4.2.1 Measure of interestingness (MoI)

The question of what is a significant enough difference in each of the mapped dimensions is not a simple problem, even with prior knowledge of the space [28, 30, 31, 142]. The difference threshold set means that only individuals with this level of difference in one of these dimensions are considered distinct enough from all other members of the population and thus a "novel" solution to the end population. Furthermore, we should consider that this level of difference might not be uniform across the entirety of any particular mapped dimension (thus resulting in varying levels of cell resolution across the space). If this were the case then the presence of smaller cells (i.e. higher resolution) in the space, for example, would increase the likelihood of those areas being explored more during the search. Moreover, random selection would be more likely to select these cells based purely on their increased prevalence in the stored population. Thus, cells at these levels of resolution would have a unreasonably large effect on the focus of the search.

Here, in order to set an appropriate cell resolution for a given unknown problem space, we seek to gain information about the space during the search itself. Ideally this should occur without additional processes slowing down the search or extensive sampling of the entire space of solutions. As previously mentioned, in MAP-Elites, mutations that lead to individuals falling within cells already known to the search (called revisits) only contribute to the optimisation of the fitness of that cell. Furthermore, the optimisation only occurs if the revisit has a higher fitness than the individual currently stored in the cell. Here, we suggest making use of these revisits, regardless of their fitness, to gain further knowledge of the space and inform our choice of the resolution of the search. This allows the search to be more accurate as it observes more samples within the space.

The measure gathered during these revisits should of course be highly dependent on the type of problem being investigated and the question being asked of the space being explored. At the minimum, this measure could be set to any measure of the behaviour of an individual that would not be well-suited to use as one of the mapped dimensions of the cell. These comprise behaviours whose relationship to the genome encoding is "non-direct"; that is, for which movement in genome space has little or no relation to movement in the given behaviour space. The extent to which the encoding is directly related to the behaviours used for cell mapping has been shown to play an important role in the effectiveness of this kind of illumination algorithm [167]. However, the selection of a measure in this way eliminates many behaviours that might still be of great interest to the search space.

In previous work, we have shown the effectiveness of using one such behavioural measure with a non-direct behaviour-to-genome relationship in the exploration of the space of network structures sharing given structural features [131]. In that work, we set the MoI within each revisited cell to be the variance in this measure, in order to determine the areas of greatest interest. By using the variances in this measure we were able to focus our search on areas of high diversity, without having to include the measure itself in the mapped

dimensions. Furthermore, as long as we assumed some relationship between the genome and the behavioural measure, we could expect at least some decrease in the variance as the resolution of cells increased. This lowered the number of possible solutions within these smaller cells, thus reducing the likelihood of the method simply "drilling down" on a few areas of high interest too quickly.

Inspired by this, here we propose to use the variance in fitness within each cell as the MoI, thus focusing our search on areas of the mapped space showing the largest effect in fitness for the smallest movement in the mapped dimensions. Practically, each cell maintains a copy of the specification of the fittest individual , along with the variance in the MoI calculated over all of the individuals sampled when the cell was visited. This adds slightly to the computational cost of each mutation compared with the original MAP-Elites, but not significantly as the variance (and all ancillary variables) can be calculated incrementally (i.e., without storing the specifications of the individuals).

Using this approach allows the diversity of fitness in the results to be increased, whilst preserving high-performing individuals via the elitism built into the MAP-Elites framework. Furthermore, the exploration is focused on the areas of the space that most affect the end fitness of the individuals. This measure also benefits from the assumed decrease in possible variability with increasing cell resolution, as reported previously in [130]. Moreover, it has the advantage of being applicable to almost all problem spaces that include an element of optimisation. Thus, it is well-suited to making the comparisons reported in Section 4.3.

A fuller discussion of other suitable MoIs is provided in Section 4.4. However, regardless of the MoI chosen, it is important to consider the speed at which the space is divided based on this MoI. Below we detail the factors used to control the rate of cell divisions and how this is best calibrated to the MoI chosen to explore the space.

4.2.2 When should cell size be changed and by how much?

In dividing the space of solutions along the chosen mapping dimensions, we are aiming to specify, using their chosen size, the importance that each of these dimensions plays in determining the MoI, as well as any variance in this importance across the dimensions. In other words, the dimensions with the smallest cells can be assumed to have the greatest effect on the MoI in those parts of the space (when flattening the view of solutions to just a 1D view of the multi-dimension space). By linking cell size to MoI, we are able to identify the areas with the highest MoI levels as being those with the highest number of overlapping cells for a single mapped dimension (see Figure 4.1).

This measure of the space is even possible without needing to change the individual amount that each dimension is reduced by during the resolution change of the selected cell (hereafter referred to as a "change event"). As such, here we use a fixed reduction (50% of the current size) for all dimensions of the cells during a change event, and focus on ensuring a good coverage of the space of possible solutions by controlling when the change events occur. Of course, in the type of unknown spaces that we wish to use this algorithm for, determining good coverage of the space is not a simple problem.

Therefore, here we propose basing the decision of when to change the resolution of the selected cells on the ratio of global exploration (the number of new cells discovered since the last change event, "Ad") to global exploitation (the number of new offspring falling within cells that have already been discovered since the last change event, "Vr"). Concretely, we use the following condition:

$$Vr > (Ro \times (Ad + LL))$$

Where Ro and LL are constants set at the start of the evolution, which are defined in more detail below. This approach can be thought of as evaluating the level of confidence we have in the coverage of the search space. Indeed, starting the search with very large cells across

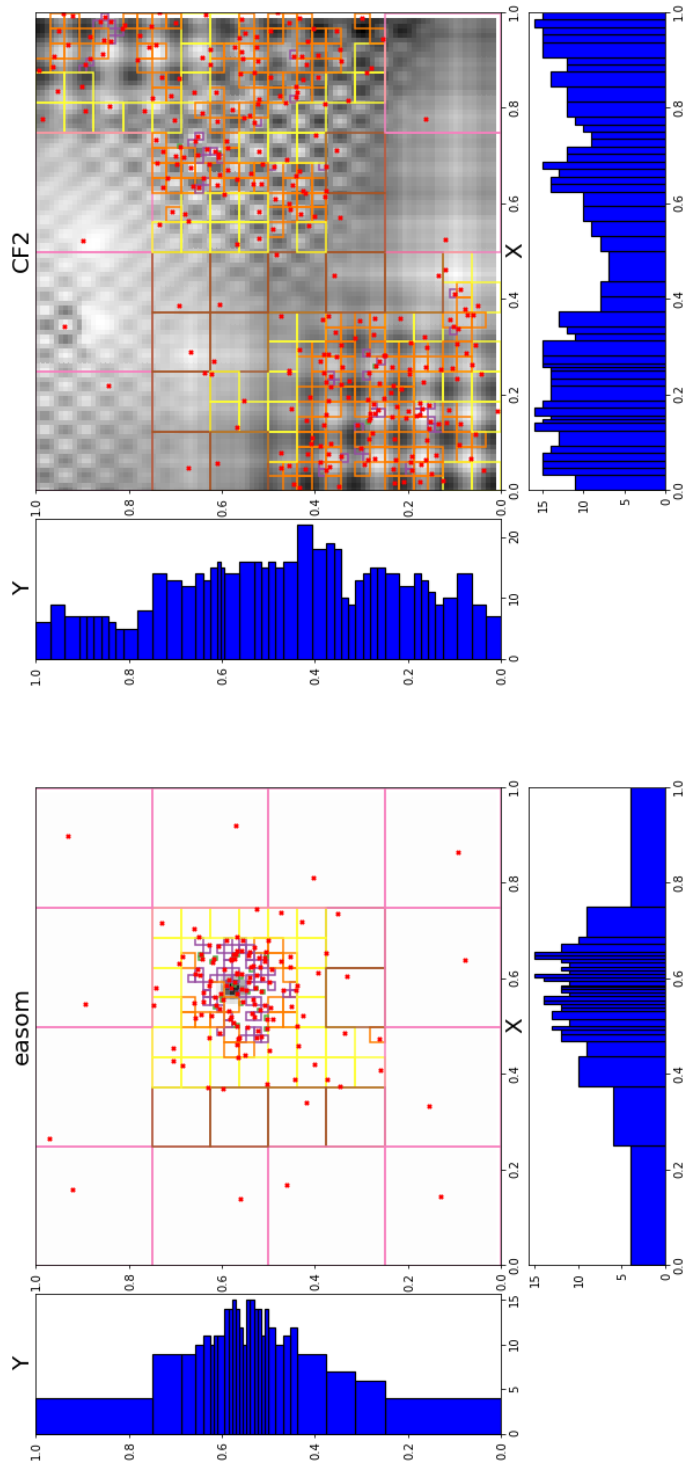


Figure 4.1: Shows the number of overlapping filled cells (i.e. cells with a found solution whose cell limit is withing the bin size) across the X (bottom most panel of each plot) and Y (left most panel of each plot) dimensions respectively. The size of each bin represents the smallest cell size seen at that point in the space. Lines show the size of the smallest cells covered in the given part of the space, for the eason (left) and CF2 (right) landscapes (details of both landscapes can be found in the Appendix 4.5). The cells in both landscapes were divided based on the level of variance in fitness (i.e., the MoI discussed in Section 4.2.1) for 2,000 iterations of the our adaptive resolution change (ARC) method.

all dimensions (thus preserving as much of the selective pressure gained from these larger cells as possible) means that when the value of Vr is greater than that of Ad, it suggests that the problem space, as defined by the current resolutions of cell across the space, is becoming saturated. Thus, we can have some confidence in our sampling of the space. Of course this balance between Vr and Ad is unlikely to be optimal for all problem landscapes or goals in the space (as characterised by the choice of MoI). Thus, we allow this balanced to be calibrated according to three factors: (1) the triggering parameter (Ro); (2) the lower limit (LL); and (3) the number of cells selected for a resolution change during each discrete change event (NC). These factors are described in detail below.

Ro specifies the extent to which Vr must exceed Ad ($+ LL$, of course) before a change event is triggered. This serves to control the overall speed at which change events occur, as well as the balance between exploration and exploration required for a change event to occur. A low value of Ro results in a significantly lower level of exploration of new cells being required to trigger change events; thus, a greater number of change events occur from the same mutation attempt. In spaces in which finding new solutions is hard (i.e., spaces in which there are few solutions in the space, or in which the solutions are sparsely distributed across the space), a high value of Ro might be desirable to account for this and to ensure thorough coverage of the space.

LL is the "lower limit" and denotes the minimum number of revisits that must have occurred before a change event is allowed. For example, if $LL = 10$ and $Ad = 0$, then the value of Vr must be at least $Ro * 10$ before a change event is triggered. LL plays a similar role to Ro, in that it affects the overall speed at which change events occur. However, rather than ensuring a certain level of exploration, it guarantees the minimum value of Vr and thus the minimum number of samples for our MoI. If LL is too low then change events can happen as soon as Vr is greater than $Ro \times Ad$, i.e. if $Ro = 1$ then this is as soon as the number of revisits is greater than Ad, such that there is no guarantee of a reasonable number of samples taken

from the space in order to obtain an accurate assessment of the MoI.

Finally, NC is the maximum number of cells to be selected for splitting during each change event. For example, with $NC = 2$, the resolutions of two cells are doubled (i.e., their sizes are halved) during each change event. These cells will those with the two highest MoI scores. It is worth noting that if the MoI score of a selected cell is zero, or if the NC value exceeds the number of cells in the current population, then that cell will not be changed and the number of cells split in the current change event will be lower than the value of NC. The NC value can thus be thought of as a measure of confidence in the values of the MoI, with a higher NC meaning that greater numbers of the high MoI cells are split more often. However, if NC is too high, it could result in the search becoming closer to a random search. In combination, these three factors can strongly influence the effectiveness of the search by focusing change events on those areas with the highest MoI. To demonstrate this, simulations using varying values of NC, LL and Ro were run on three landscapes of varying behavioural ruggedness (as defined by the gradient of behavioural change in the landscapes) and numbers of areas of interest (as defined by the number of global optima for the MoI). All solutions in the spaces were assumed to have the same fitness for the purpose of cell elitism, and thus there was no optimisation at work in these examples. The MoI used for all landscapes considered was the level of variance in the height of the "behaviour" within the landscape (for a mathematical and graphical description of the landscapes, see Appendix 4.5.1). To map different scenarios, we chose the Perm function (Figure 4.4) as an example of low-behavioural ruggedness, with four distinct areas of interest and very wide gradients towards each optimum. In contrast, the De Jong and Easom landscapes have very high behavioural ruggedness in all areas of interest, with Easom having a very low number of areas (one area of interest) and De Jong showing the highest number of the three landscapes (25 areas of interest).

Across all landscapes, with low LL and Ro (here $LL = 0$ and $Ro = 1$), greater numbers of

cells are created, as expected. This is most prominent in the Perm landscapes (Figure 4.4), which show the largest behavioural gradients leading to high fitness areas and thus gain the most information from revisits. Conversely, with low LL and Ro in the Easom landscape, we see a greatly reduced effect on the placement of cells, due to the limited amount of information available in that space. Similarly, the size of the additional cells is more affected in the Perm and De Jong landscapes, with significantly smaller cells in the low LL and Ro conditions. This is to be expected with these landscapes, since De Jong has multiple areas of interest and although Perms has only a few areas of interest, they are very wide. Thus, it is only in the Easom space that we see drilling down further than this (to 0.008, i.e. five change events). For this landscape, in the low LL and Ro conditions, the choice of cells split was sub-optimal compared with that in the high LL and Ro conditions ($LL = 10$ and $Ro = 4$). These effects are reversed when high and low NC are examined across all three landscapes, with high NC showing a similar decrease in the number of solutions to that seen in the low LL and Ro conditions, albeit with a greatly reduced effect.

This balance of revisits (which provide the information that determines the choice and speed of change events, as well as any optimisation of fitness within a cell) and the speed at which solutions are found within the space will of course differ. This largely depends on the MoI used and its relationship to the behaviours and/or genomes being mapped, which is assumed to be unknown for our purposes. However, within the three landscapes shown here, it could be suggested that an informed balanced average of these 3 features (here set to $NC = 2$, $LL = 3$, and $Ro = 2$ in the centre panels of Figures 4.2, 4.4, and 4.3) would of course give the best compromise between revisiting and the speed at which solutions are found. However in contrast, high revisiting (with high LL and Ro and/or lower NC) is less likely to have a deleterious effect on the search as a whole, even at the cost of the speed of solutions, than the reverse (low revisiting and high NC).

As discussed above, changing the resolution of the cells in the space helps to focus the

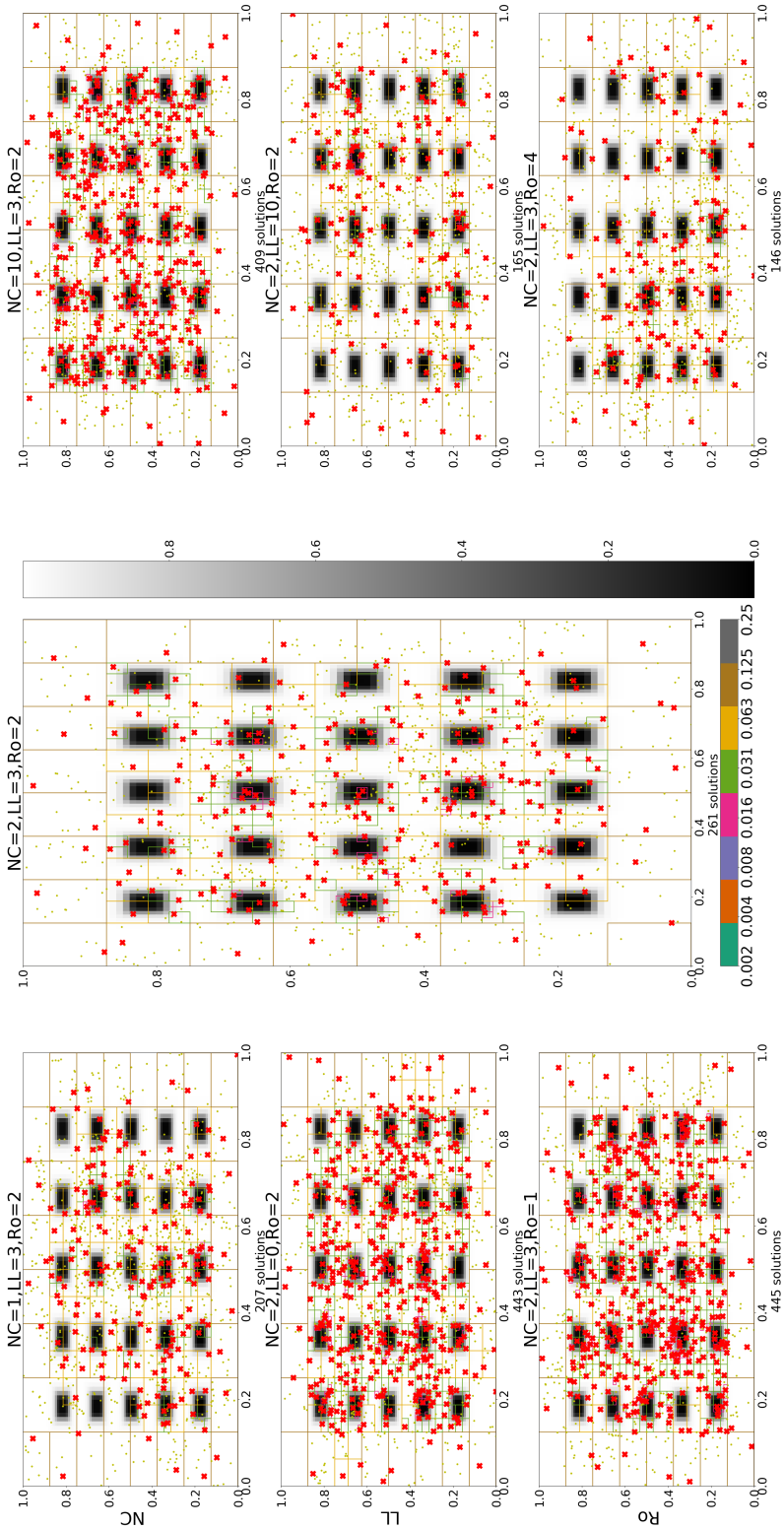


Figure 4.2: Divisions of the 2D De Jong (De) landscape (see Appendix 4.20) after 2,000 iterations. Each panel shows one of the seven conditions: High, mid, and low values for each of the three values of NC (top row), LL (middle row), and Ro (bottom row), with the centre panel showing the mid values for each. The default value used in each is $NC = 2$, $LL = 3$, and $Ro = 2$. The height of the landscape (bar to the right of the middle panel), colour-coded cell size (bar at the bottom of the middle panel), revisits (yellow dots), and the elite solution stored at the time of plotting (red cross) are shown. However, due to the method used for plotting, only those cells that were filled at the time of plotting have their cell size shown.

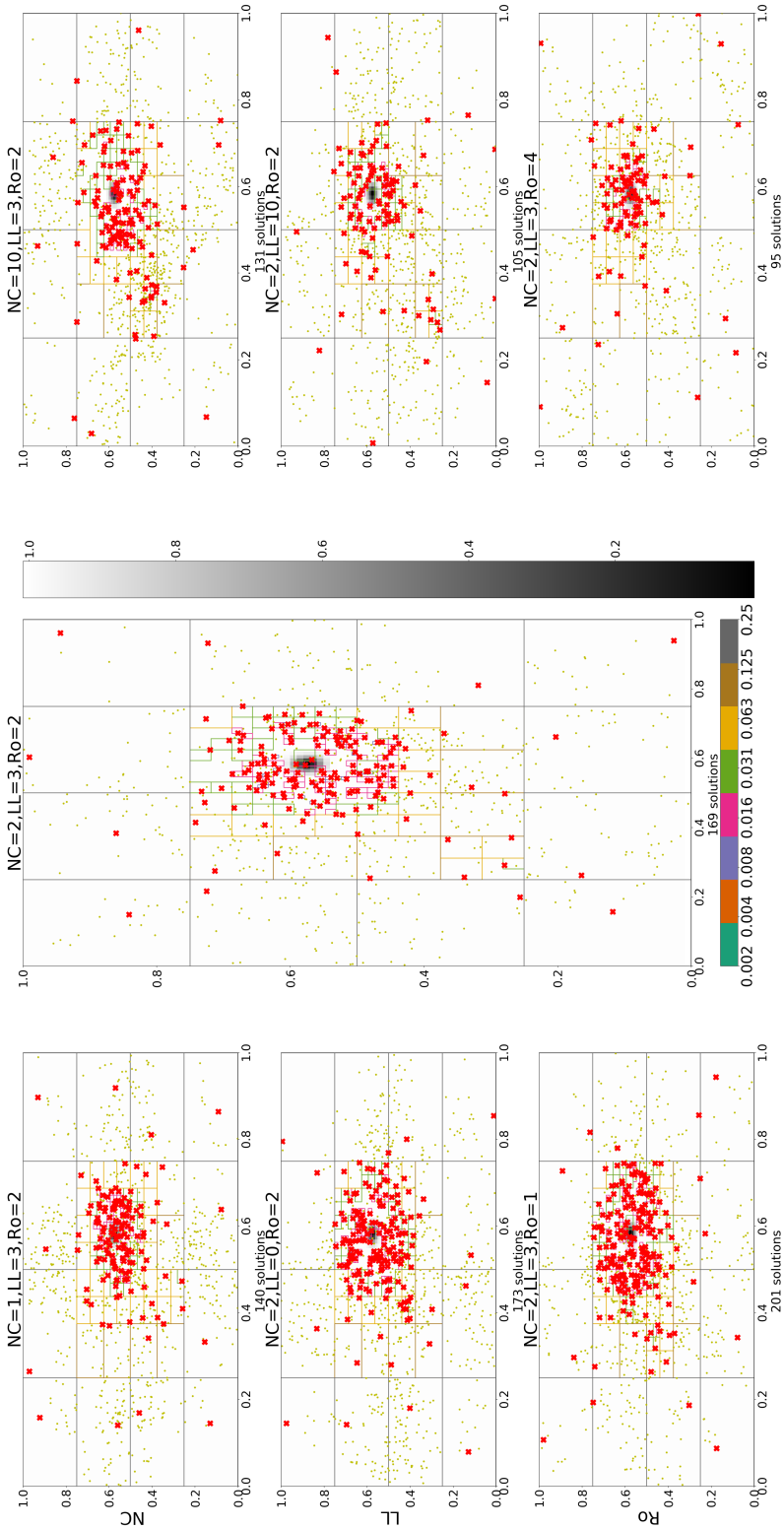


Figure 4.3: Divisions of the 2D Easom landscape (see Appendix 4.21) after 2,000 iterations. Each panel shows one of the seven conditions: High, mid, and low values for each of the three values of NC (top row), LL (middle row) and Ro (bottom row), with the centre panel showing the mid values for each. The default value used in each is $NC = 2$, $LL = 3$, and $Ro = 2$. The height of the landscape (bar to the right of the middle panel), colour-coded cell size (bar at the bottom of the middle panel), revisits (yellow dots), and the elite solution stored at the time of plotting (red cross) are shown. However, due to the method used for plotting, only those cells that were filled at the time of plotting have their cell size shown.

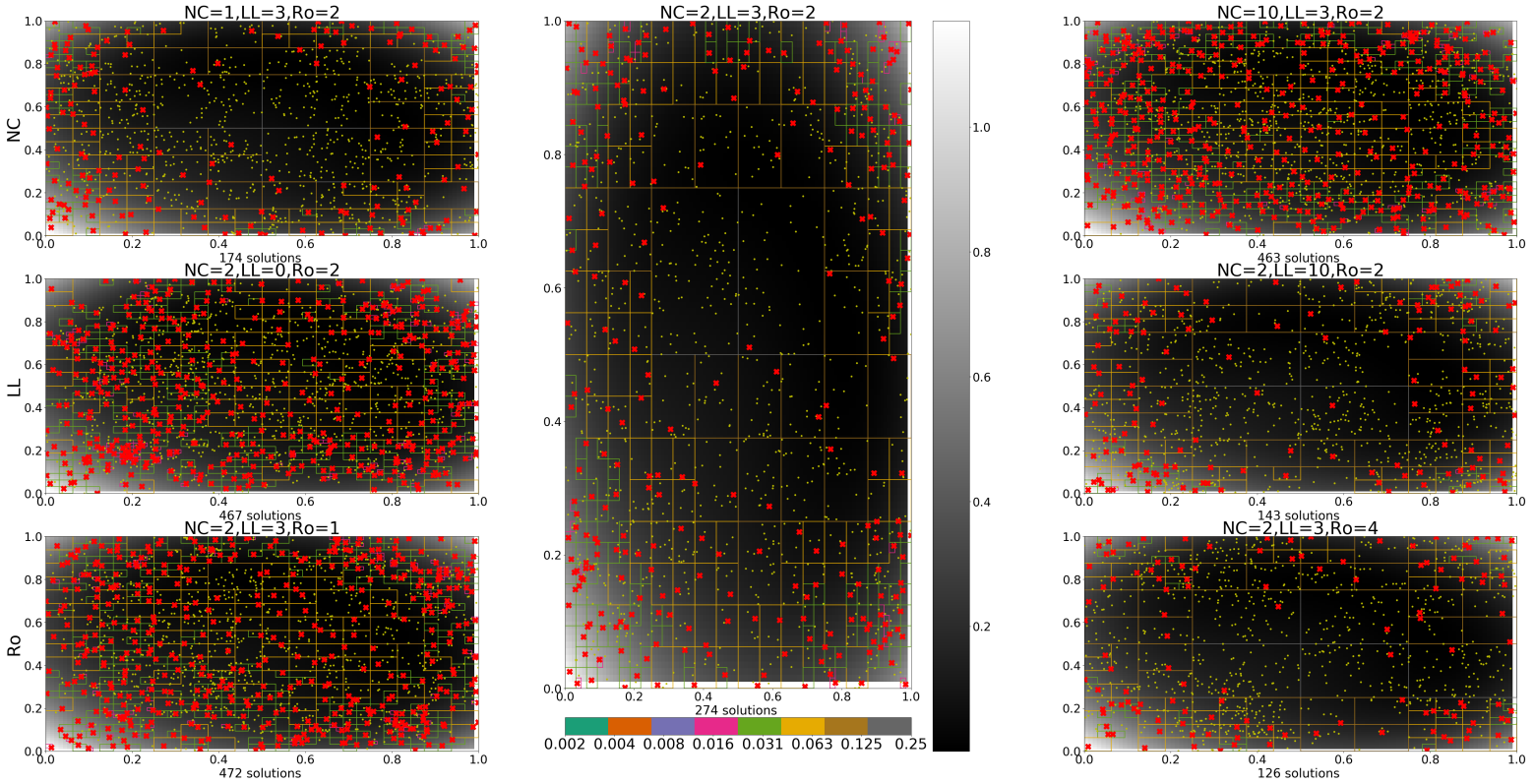


Figure 4.4: Divisions of the 2D Perm landscape (see Appendix 4.22) after 2,000 iterations. Each panel shows one of the seven conditions: High, mid, and low values for each of the three values of NC (top row), LL (mid row), and Ro (bottom row), with the centre panel showing the mid values for each. The default value used in each is $NC = 2$, $LL = 3$, and $Ro = 2$. The height of the landscape (bar to the right of the middle panel), colour-coded cell size (bar at the bottom of the middle panel), revisits (yellow dots), and the elite solution stored at the time of plotting (red cross) are shown. However, due to the method used for plotting, only those cells that were filled at the time of plotting have their cell size shown.

search towards the areas of greatest interest and increases the exploration of those spaces; however, this is not the only factor to consider. In the next section, we discuss the effect of adding an adaptive mutation rate to tune the movement of the search through the space as the cell resolution changes.

4.2.3 Adaptive mutation range

The way in which the search moves around the space of solutions can have a massive effect on the kinds of solutions found [168, 181, 64, 148, 166]. This is especially true for the balance of exploration vs exploitation, with coarse movements normally facilitating exploration, while more fine-grained movement contributes to exploitation. When space of stored solutions is redefined to reflect the areas of greatest interest for a given investigation (e.g., by changing the cell resolution in the mapped space), it is desirable for the movement in these regions to also reflect this varying level of interest. Without such a change in the movement of the search, the desired focusing of the search is unlikely to result from the redefinition of the space of stored solutions. For example, when adaptive mutations are not used (referred to as the "no adaptive used" condition), which keeps mutations in a range of 0.125 to 0.5 (regardless of the parent cell size) throughout the search, there is significantly less revisiting of the areas of high MoI than when adaptive methods are used. This is particularly clear in the Easom landscapes (see right-most panel of the middle row of Figure 4.5) and also results in a significantly lower number of solutions. This is because the number of change events decreases, which can be seen in all of the landscapes tested. As such, we propose the use of an adaptive mutation method that allows the size of mutations within the search space to progress from coarse to fine based on the level of interestingness shown by each newly defined region of the space. The concept of using an adaptive mutation size in response to increasing knowledge of the space is well established in work that seeks to control the trade-off between exploration and exploitation in GAs [45], including within the

MAP-Elites framework (e.g., see [126]). Here we implement this concept by proportionally linking the range of the mutation to the resolution of the parent cell used to create the mutated offspring. That is, each of the selected genes in the genome is changed by $\pm$ a value randomly selected from a range related to the resolution of the parent cell (e.g., $cellsize/2$ to $cellsize \times 2$. This means that at the start of the search when all of the cells are large and the focus should be on increasing coverage of the space, mutations lead to more of a global search. Then, as the average cell resolution increases, the search tends towards more of a local search around areas of high MoI. Note that this linking of the mutation range to cell resolution does assume that a movement in the genome space will result in a similarly sized movement in the behaviour space (i.e., the space being mapped via the cells). However, as shown by other studies (e.g., [167]), the more "direct" the encoding used with MAP-Elites, the more effective the results are, both in terms of diversity and quality (fitness).

It is also worth noting that the effects of this change in mutation range will be more or less appropriate depending on the difficulty of the problem landscape explored. For example, when the gradient of fitness is spread over a wider proportion of the space (and thus easier to find and follow during the search), such as in the Perm landscape (see Figure 4.5), the focusing of the search is greatly reduced compared to that in harder landscapes such as Easom or even De.

For example, when using the "in cell" mutation range, there is a clear focus of the revisiting on areas of the space already seen. Such mutations correspond to the range [cell size/2 to cell size], which gives a high likelihood (approximately two-thirds of the time, as evaluated from simulations using normally distributed mutations from this range) of most offspring being placed somewhere within the parent cell. This is particularly clear in the more difficult landscapes such as De, in which the revisiting clearly focuses on areas of the space already seen. This is true to the point that some areas of the space that are not seen early on in the search are completely ignored (see the right-hand bottom corner panel of Figure 4.5).

Worse still, in a landscape such as Easom in which there is little difference in the fitness gradient across the majority of the space, this high revisiting of areas can result in an over-representation of any small difference identified early on, because of the continuous resampling of that area of the space.

In contrast, when the "out of cell" range is examined (in which the range of mutation is [cell size to cell size*2], such that all mutations land outside the parent cell), the level of exploration is much higher. However, because this means that revisiting is only achievable via mutations from neighbouring cells, there are still a lot of "wasted" mutations that do not result in an improved solution (i.e., the level of optimisation within each cell might be limited). A combination of these two implementations, which uses each of the above implementations randomly 50% of the time (referred to as the "0.5" condition hereafter), results in less "wasteful" revisiting of the space than when a mutation range that covers both ranges is used (referred to as the "full range" condition hereafter). It is worth noting that due to the chance of the "in cell" condition leading to mutations outside the cell, the "0.5" condition is slightly more focused towards mutations outside the parent cell (with around 65% of the mutations resulting in movement outside the parent cell). However, this still shows that limiting movement to either within or outside the parent cell is more effective than merely increasing the size of the mutation range to allow for both of these possibilities. This would thus appear to suggest that the choice of the adaptive mutation range should be carefully considered in relation to the level of optimisation vs exploration desired from the search. Here we have chosen to focus the majority of our work on high exploration over optimisation, the former being the less commonly investigated focus. Thus in the results section below, the "out of cell" condition was used in all but the hexapod experiment, which used the "0.5" condition instead as there is a greater need for optimisation in that problem.

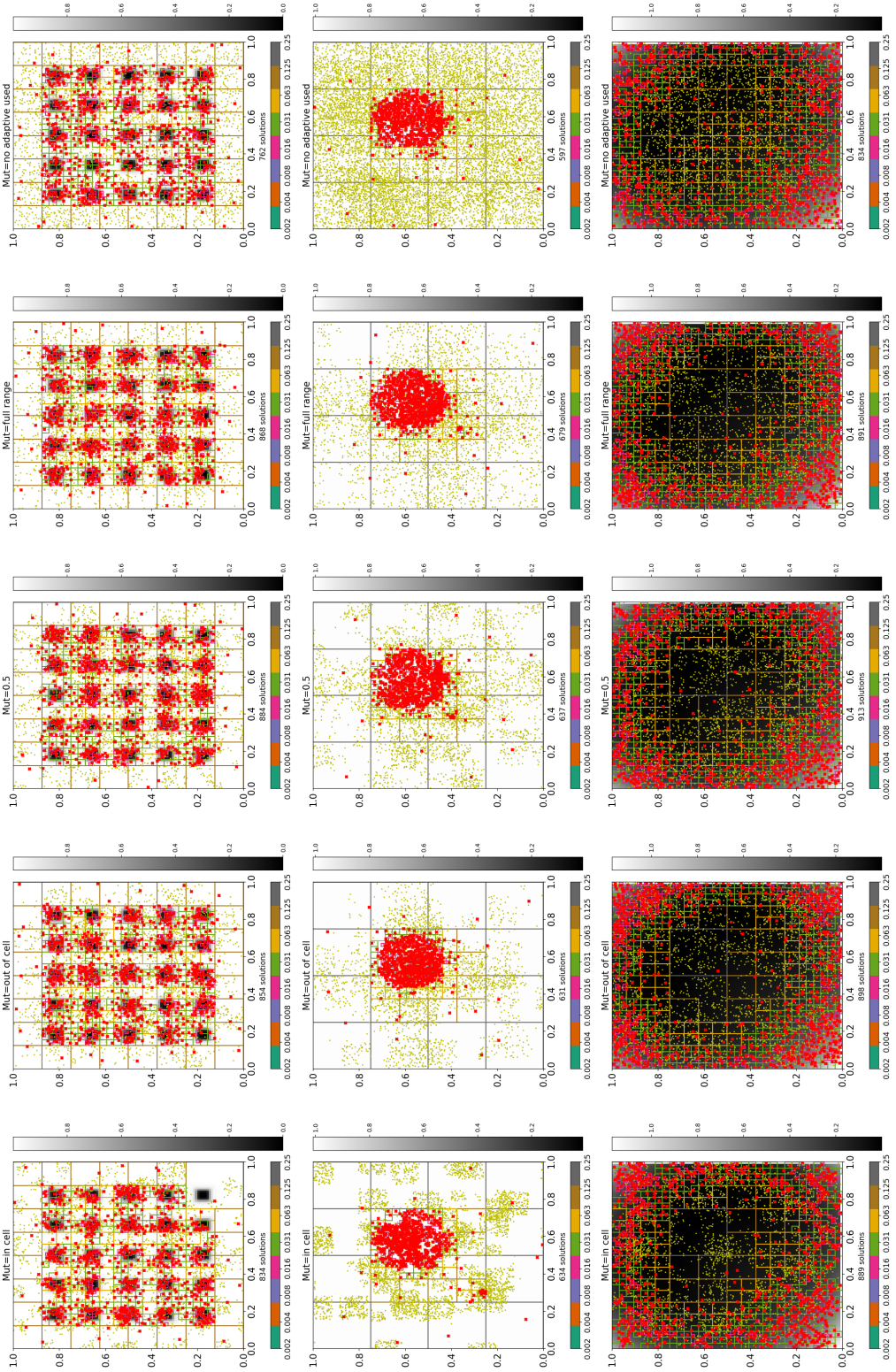


Figure 4.5: Division of the 2D landscapes after 500 change events for the De (top), Easom (middle), and Perm0db (bottom) landscapes (see Appendix Figures 4.20, 4.21, and 4.22). Red crosses indicate the location of each solution in the end population, while yellow dots indicate "revisiting" mutations.

4.3 Results

In order to demonstrate the effectiveness of our proposed ARC method, we compare its performance with the original, fixed cell resolution, MAP-Elites method [118] in three distinct scenarios: (1) three 2D-niching landscapes [95]; (2) an exploration of the space of network structures sharing only a fixed level of global clustering and degree distribution [131]; and (3) a hexapod walking controller simulation [174]. Each of these scenarios is detailed below.

4.3.1 Niching landscapes

In population-based metaheuristics such as MAP-Elites (and in all GAs), there is a need to maintain a good range of diversity in the population of solutions, in order to avoid problems such as convergence to a local optimum. Furthermore, when considering multimodal problem spaces (i.e., spaces with no single optimum [44]), without good diversity in the population of solutions, methods often find it hard to return more than one distinct high-fitness solution. As previously mentioned, one of the major advantages attributed to the MAP-Elites method is its ability to find multiple high-fitness solutions from across the problem space. The problem of identifying all niches within a space, here characterised by distinct global optima or "niches", is of particular relevance to measuring how well the method performs at identifying all areas of interest within a problem space. Note that this is not to say that we are attempting to fully cover the space of all possible solutions, but merely that all spaces of interest are covered by the end population. A number of benchmark functions have been proposed to evaluate the effectiveness of a given method in finding niches within multimodal composition functions with several global optima. One of the most widely used benchmarks is the CEC'2013 [95]. Below, we compare the effectiveness of both MAP-Elites and the ARC method using five independent runs of the following conditions

(see Appendix 4.5.1 for the parameters used in each condition): either large cell size (0.5, meaning a total of 4 starting cells) or small cell size (0.25, meaning a total of 8 starting cells) using three of the 2D composition functions provided in the CEC'2013 test suite, namely composition function 1 (CF1), composition function 2 (CF2), and composition function 3 (CF3). These landscapes were all normalised such that their parameter space ranged from 0 to 1. They all have a fixed number of global optima (six for cf1 and cf3, and eight for cf2), which is below the number of starting cells in the large condition (which has a starting cell size of 0.5), but above (or equal) to the number of starting cells in the small condition (which has a starting cell size of 0.25). When knowledge of these spaces is not available to inform the choice of starting cell size in a MAP-Elites implementation, it is reasonable to assume that the focus on preserving the selective pressure within cells would prompt the favouring of larger starting cell size condition (see section 4.2) over the small condition.

In the larger starting cell size condition, we can see that there is no way that the standard MAP-Elites method can cover all niches in any of the benchmark landscapes (see the left-hand column of Figure 4.6). Since it has only one elite to summarise the entirety of these large cells, MAP-Elites proves to be highly effective at optimising solutions that find the global optimum within each cell. Overall, it finds global optima very close to the actual peaks in all of the landscapes (note the 0.01 accuracy in Figure 4.8). However, when there are multiple global optima within the same cell, we observe a process of cycling from one optimum to another within the cell. This is seen most clearly in the CF1 landscapes, in which there is repeated movement from one peak to the other between the 200th and 20,000th mutation (panels a and b respectively in Figure 4.6).

In the smaller starting cell size condition, in which there are enough cells to cover all of the global optima within each landscape, we observe that MAP-Elites is much more effective in finding all global optima. This can be seen by the higher mean percentage of niches found in all of the landscapes analysed (see Figure 4.8). However, these optima are not as close to

their potential "peak" fitness, instead showing the reduced level of optimisation expected from using smaller cells. Indeed, the use of smaller cells means that there are fewer solutions in each cell in competition with each other. This therefore lowers the overall level of competition in the search (e.g., note the 0.01 accuracy level in Figure 4.8). Nevertheless, there are still many cases in which the elites of two neighbouring cells converge to the same global optimum; for example, in the following cases: the two elites around $x = 0.9$, $y = 0.75$ in the CF3 landscape (left-most panel of plot a in Figure 4.7); the elites around $x = 0.5$, $y = 0.9$ and $x = 0.9$, $y = 0.75$ in the CF2 landscape (centre-top panel of plot a in Figure 4.7); and the elites at $x = 0.9$, $y = 0.75$ in the CF1 landscape (right-most top panel of plot a in Figure 4.7).

In contrast, when the ARC method is used, we observe a focusing of the smaller cells around the areas of highest MoI (i.e., areas of high variance in fitness) across all three landscapes, with larger cells (meaning a lower number of change events) in the areas of low MoI. Furthermore, the distributions of the sizes of these cells remain very similar regardless of the starting cell size, with most of the differences being between the larger cells in the 0.5 starting cell size condition. This would suggest that, given enough time, despite the larger starting cell size in the 0.5 condition (corresponding to insufficient starting cells to cover all niches in the space), the ARC method is able to find the same number of global optima to the same level of accuracy as in the small cell size condition.

However, there is a clear difference in performance between the two conditions when the ARC method is used for all of the landscapes. While not as pronounced as the difference obtained when the starting cell size conditions are compared after using MAP-Elites, there is still a higher mean percentage of niches found with the smaller starting cell size condition. However, unlike in the MAP-Elites examples, there seems to be a greater level of accuracy gained from the smaller starting cell size in all but the CF3 landscapes. This difference in the CF3 landscape is likely due to it having the widest distance between niches, which

means that the larger starting cell size may yield a greater chance of covering the space before the search begins to focus on the identified areas of greatest interest.

These results are of course limited to 2D, but do serve to demonstrate, in a readily visible space, the kind of limitations that an uninformed MAP-Elites method faces in capturing the full range of global optima in an unknown feature space. It is worth pointing out that it is not the primary function of MAP-Elites to find all global optima (i.e., it is not intended to be a niching algorithm). Rather, it merely attempts to approximate the niches within the landscape in order to maximise coverage of the space of generated solutions. As such, although these results clearly demonstrate the limited ability of MAP-Elites to represent the full range of solutions within the explored space, the full benefit of the MAP-Elites method might not be best summarised by this kind of low-dimensional problem space.

Next, we demonstrate the use of the ARC method on a much more complex problem space, exploring the space of possible network structures sharing a fixed level of global clustering and degree distribution.

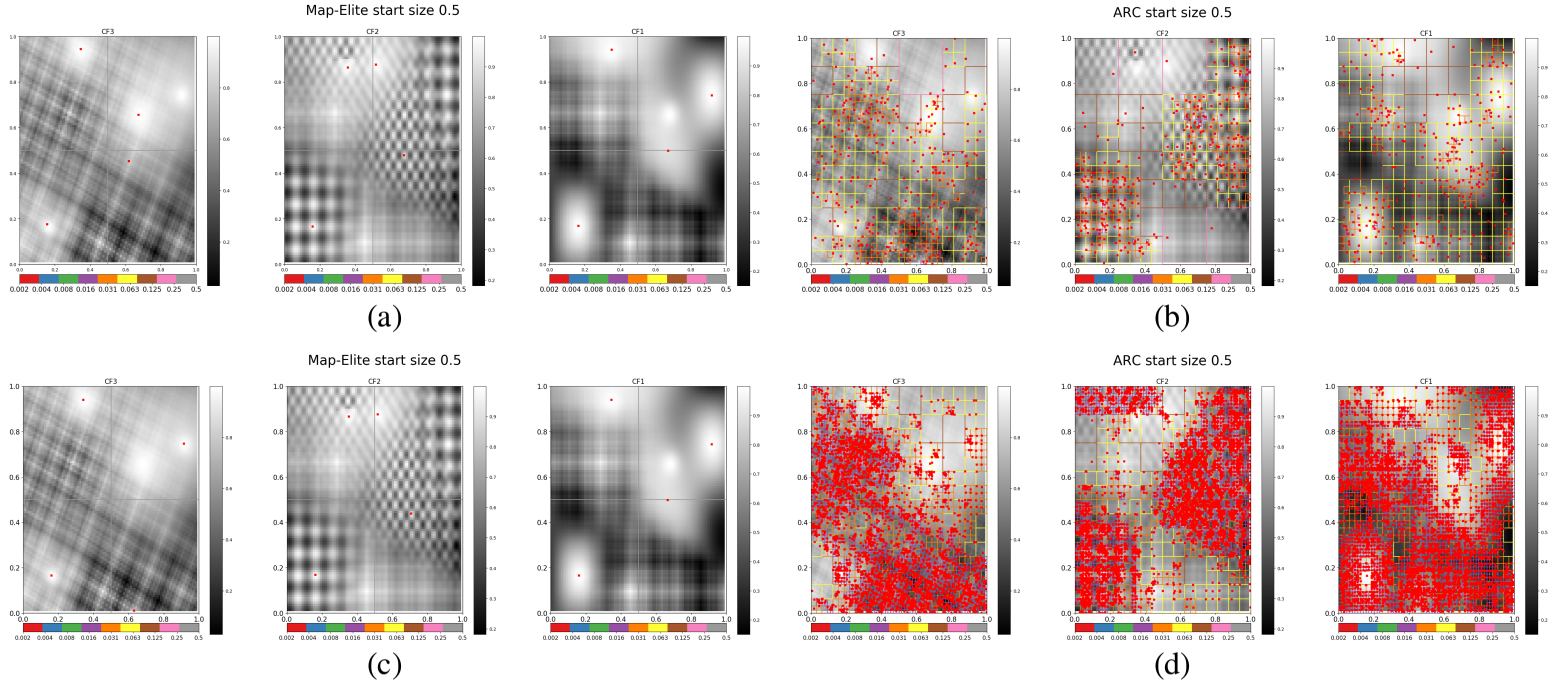


Figure 4.6: Division of the 2D landscapes after 2,000 mutations (top row, panels a and b) or 200,000 mutations (bottom row, panels c and d) for the ARC (right column, panels b and d) and MAP-Elites (left column, panels a and c) methods, using the 0.5 (large) starting cell size condition. Red crosses indicate the location of each solution in the end population. Left to right: the landscapes used in each condition are CF3, CF2, and CF1, respectively.

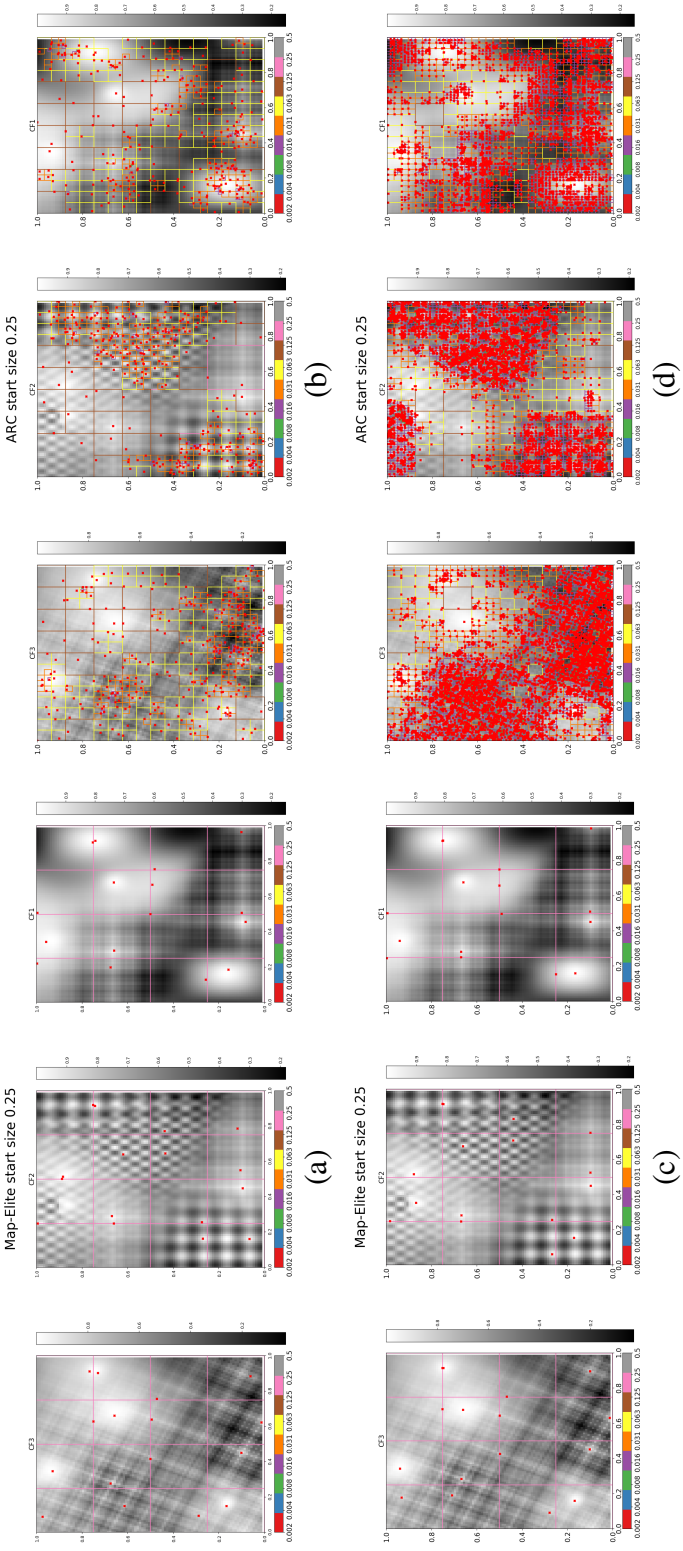


Figure 4.7: Division of the 2D landscapes after 2,000 mutations (top row, panels a and b) and 200,000 mutations (bottom row, panels c and d) for the ARC (right column, panels b and d) and MAP-Elites (left column, panels a and c) methods, using the 0.25 (small) starting cell size condition. Red crosses indicate the location of each solution in the end population. Left to right: the landscapes used in each condition are CF3, CF2, and CF1, respectively.

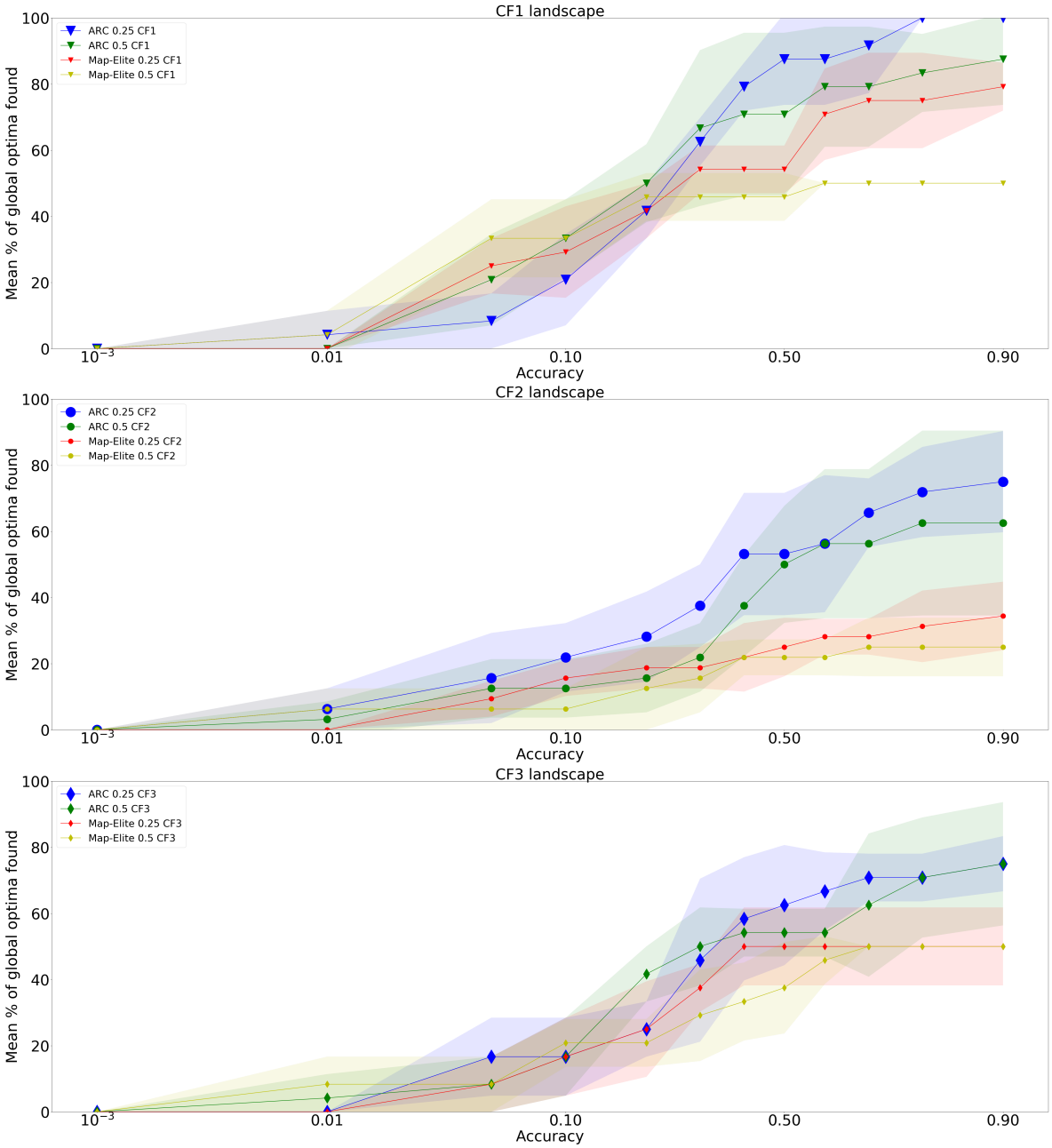


Figure 4.8: Mean and standard deviation (shown with the shaded areas) of the percentage of global optima found with different levels of accuracy (i.e., the maximum difference tolerated from the true optima still classified as indicating that one of the optima has been found), after 200,000 mutations for each of the following conditions: ARC with a starting cell size of 0.25 (blue); ARC with a starting cell size of 0.5 (green); MAP-Elites with a starting cell size of 0.25 (red); and MAP-Elites with a starting cell size of 0.5 (yellow).

4.3.2 Exploration of network structures

In network science, the relationships between interconnected parts of a system are typically represented by networks consisting of nodes and edges [119]. The way in which these nodes and edges are connected is referred to as the network structure. Many real-world networks that have similar behavioural features when the same dynamics are run on them tend to share similar structural features [121, 19, 115]. For example, many real-world networks that are intended to model behaviours such as effects on social media and/or the spread of disease [182, 144] have been observed to show specific features. These include a particular ratio of closed triangles to open triangles + closed triangles (referred to as the global clustering coefficient) and distribution of edges for each node (referred to as the degree distribution), both of which have been covered in Section 1.2 of this thesis.

As such, these two structural features (global clustering coefficient and degree distribution) are often used as constraints in the construction of null models [150, 120, 159, 108]; that is, for the construction of networks sharing only these two structural features, which allows for fair testing of their interdependence with the dynamic being investigated.

The space of all possible network configurations is not only very large but also complex, in that not all network configurations will lead to networks with the correct structural features. This makes the exploration of this space a non-trivial task without the use of deterministic "rewiring" methods, which have been shown to be incapable of covering the full range of structural diversity in the space [131].

Here we provide new insights by revisiting earlier work using the ARC method in order to explore the space of networks that share only a set global clustering coefficient and degree distribution (see Chapter 3). This space is defined by the encoding of each network structure via the population counts of each subgraph (a small network structure within larger networks that are arbitrary in their structure and size [71]: see examples in Figure 2.1) in an arbitrarily

chosen family of subgraphs (here △, □, ◹, ⊠, and ⬠).

Mutations were performed using the Diophantine method developed in [131] (see Section 3.3.2 of Chapter 3), which guarantees that any mutation to any subgraph does not lead to a change from the global clustering coefficient and degree distribution of the parents. This results in movement across genomes with the desired global clustering coefficient and degree distribution (hereafter referred to as "valid" networks).

The mapped dimensions were linked to the genome and as such consisted of five dimensions (△, □, ◹, ⊠, and ⬠). As movement within the space is limited to valid networks, there is no need for a fitness score to be used in this evolution. Thus, the first solution found in each cell remains the stored solution for that cell from then onward. Revisiting cells already filled thus only adds to the MoI achieved over five realisations of subgraph encoding using the cardinality matching algorithm [151], where the MoI was set to be the variance in mean betweenness centrality (BC; defined as the mean number of shortest paths passing through each node in the network [52]). This MoI focuses on local level changes to the network structure, meaning that we would expect the areas of the space showing the highest MoI to be around subgraph populations which have a high variability in possible arrangements of subgraph counts.

These factors lead to a complex space in which the interconnected nature of the genome (resulting from the Diophantine mutation, which means that any change to one genome will be accounted for in the others in order to maintain valid solutions) makes it likely that many unknown regions of the space might not have any valid solutions at all.

Below we compare the performance of the ARC and MAP-Elites methods in exploring the space of networks sharing a global clustering coefficient of 0.1 and a regular degree distribution $K = 7$ (meaning that all nodes in the network must be connected to exactly seven other nodes).

For the ARC method, the LL, NC and Ro variables were set to LL=3, NC=2 and Ro=2,

in keeping with the values used in [131]. This ensures a focus on exploration over exploitation of the space, given that fitness optimisation is not required in this problem space. The same starting population was used for both the ARC and MAP-Elites methods in each cell size condition. The mutation size for MAP-Elites was fixed at the same size as would be used by the cells of the same size in the ARC method (i.e., a larger mutation size for the larger starting cell size, and a smaller mutation size for the smaller starting cell size). We assessed the "coverage" of each of the conditions, which is defined as in [167] as the number of cells (of a fixed size) that are filled (i.e., visited at least once during the search) divided by the theoretical total number of cells (i.e., the number of cells possible in the space given the size of cells used, the number of dimensions, and the boundary given the space used for mapping). In order to perform a fair comparison of the coverage, we used the same storage restrictions for both methods (i.e., a fixed cell-sized grid, in which only the solution with the highest fitness is stored; that is, the first one seen in the case of the network example), such that a coverage score of 1.0 would mean that all possible cells in the space have been filled.

In terms of the coverage of the space of possible cells (Figure 4.9), there is no difference between the ARC and MAP-Elites methods for the 64 cell size condition. However, there does seem to be some difference for the 32 cell size condition, with a slightly greater coverage by the ARC method after 3,500 generations. This difference is accompanied by an increase in the standard deviation of the coverage (shown by the shaded areas in Figure 4.9) of the MAP-Elites results. These have the same number of generations as the ARC results, so this difference could be the result of a grater consistency in the ARC coverage of the population, rather than a significantly greater coverage score. Furthermore, the standard deviation of the coverage for the ARC method using the largest starting cell size condition (64) was much greater than that of the MAP-Elites method when they were first examined (after the first 10 generations). This might indicate that the plateau seen after 500

generations in both the MAP-Elites and ARC methods might just be the maximum coverage possible in the space. This is further confirmed when we examine the coverage score of all of the conditions when their populations are treated as having the largest cell size (64) (see the central plot in Figure 4.9). Here we see that all conditions, regardless of starting cell size, reach the same plateau. This is likely the result of a saturation of the space of valid solutions, and a higher coverage of the space with this size of cell might not be possible in this space. That considered, when we examine all of the conditions with a finer cell size (16), we can see that all of the ARC conditions show significantly greater coverage compared with even the smallest starting cell size condition of MAP-Elites (see panel c in Figure 4.9). Furthermore, even when the two methods are compared at their original starting cell sizes (panels b and d in Figure 4.9), we can see that the ARC conditions are able to achieve a greater coverage of the space than the MAP-Elites conditions.

That said, this coverage measure does not really say anything about the quality of the coverage of the space (i.e., which areas of the mapped dimensions were covered and which areas were not). Examination of the distributions of these cells over the individual dimensions (△, □, ◪, ⊠, and ⬠) (see Figures 4.10, 4.11 and 4.12, which are in the same style as Figure 4.1 only with a greater number of mapped dimensions) shows a significant difference in the range of solutions found by the ARC (bottom row) and MAP-Elites (top row) methods. The ARC method conditions show a matching or larger range for all subgraphs in all starting cell size conditions, with the exception of ◪ in the 16 starting cell size condition (although there is a gap in the range covered by MAP-Elite). This suggests that the ARC method results in cells placed more widely across the space of solutions, which was also observed in the greater coverage of ARC conditions in the largest cell size results (panel a in Figure 4.9). Furthermore, the increased ranges of solutions found by the ARC conditions show no gaps, unlike the MAP-Elites conditions, in which at least one gap can be seen in every starting cell size condition. Examination of the effect of the starting cell size

also shows that within MAP-Elites conditions, peaks across the subgraphs were more common and more pronounced.This is likely due to the decreasing cell size resulting in a smaller size of mutations (large mutation sizes are traditionally linked to greater exploration of the space), such that there is a greater focus of the search with decreasing cell size. In contrast, there is little effect of the starting cell size on the range and peak of solutions found with the ARC method across all of the starting cell sizes.

These results indicate that the ARC method has a better ability to represent/map the space of network solutions even when the appropriate starting cell size is unknown. However, this problem space does not fully take advantage of the elitism of MAP-Elites, in that all solutions in a cell have equal fitness. This means that once a cell has been filled, any further revisiting of that cell contributes nothing to the search and can be thought of as a wasted mutation. Furthermore, the use of the genome space as the sole mapped dimension does not take full advantage of the kind of unique selective pressures informing the movement of the search that could be gained by adding more unrelated mapped dimensions. Therefore, in the next section, we compare the ARC and MAP-Elites methods in an optimisation task that is well known to have the benefit of using a mapping space less related to the genome—the hexapod walking task [29].

4.3.3 Hexapod walking experiment

Traditionally, the space used for mapping cells in MAP-Elites is kept distinct from the genome encoding. This is often implemented by using one or more features of interest that describe the possible behaviours of individuals over their lifetime [141, 38], which are referred to as the behaviour space. This encouragement of diversity within a behaviour space, rather than over the genome space, has been shown to be beneficial by experimental results in several separate domains [39, 116]. Investigating how novelty diversity methods (such as MAP-Elites) can be used to yield a population of high-fitness individuals that also

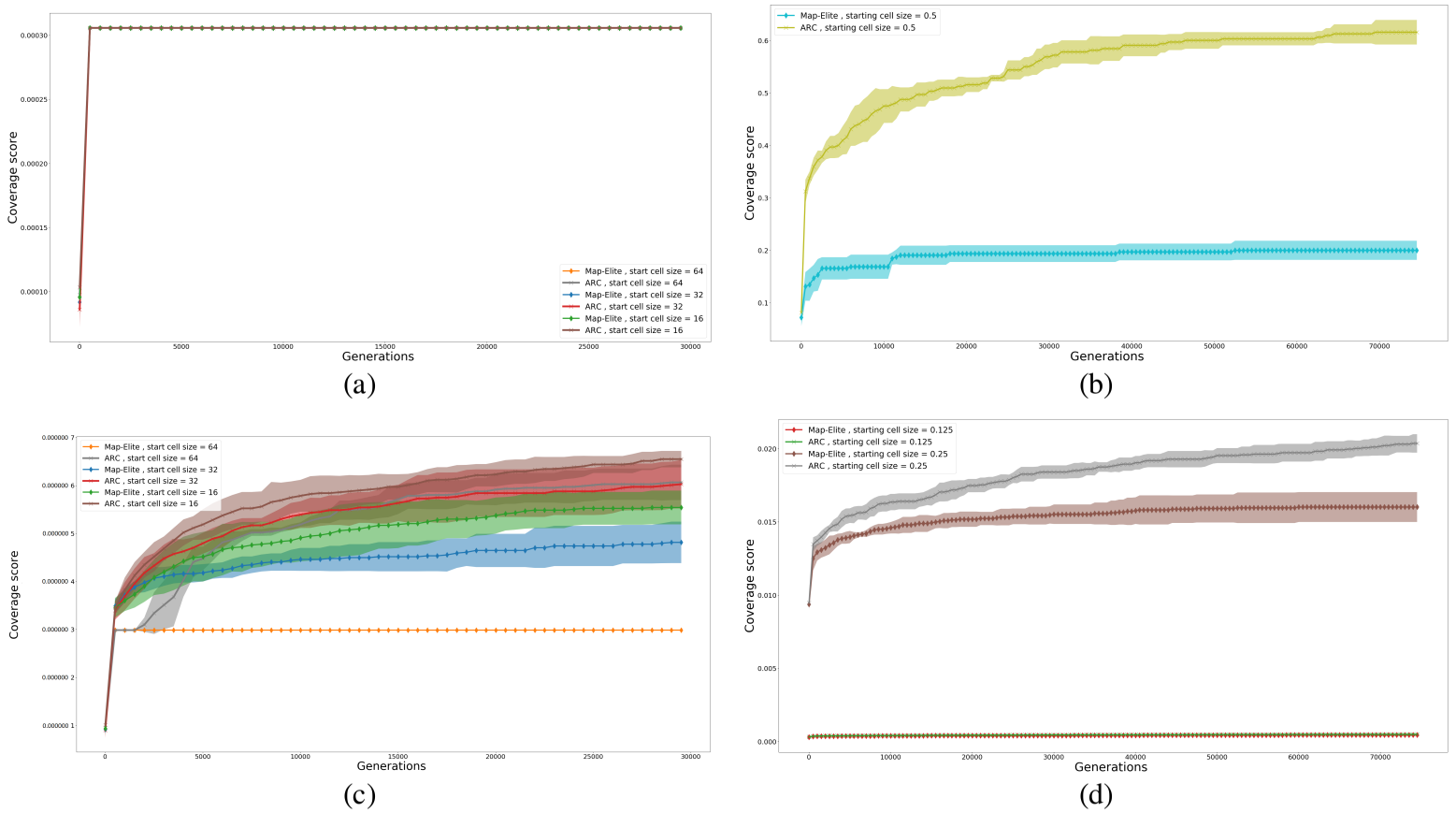


Figure 4.9: Shows the coverage (i.e., the total number of cells found divided by the total possible number of cells for that starting cell size) from the population after 30,000 iterations over five independent runs of each condition. Panel (a) shows the coverage when all conditions were compared using the large cell size (64). Panel (c) shows the coverage when all conditions were compared at the smaller cell size (16). Panels (b) and (d) (in the right-most column of panels) show the coverage when the cell size used for calculating the coverage was matched to the starting cell size of the conditions shown.

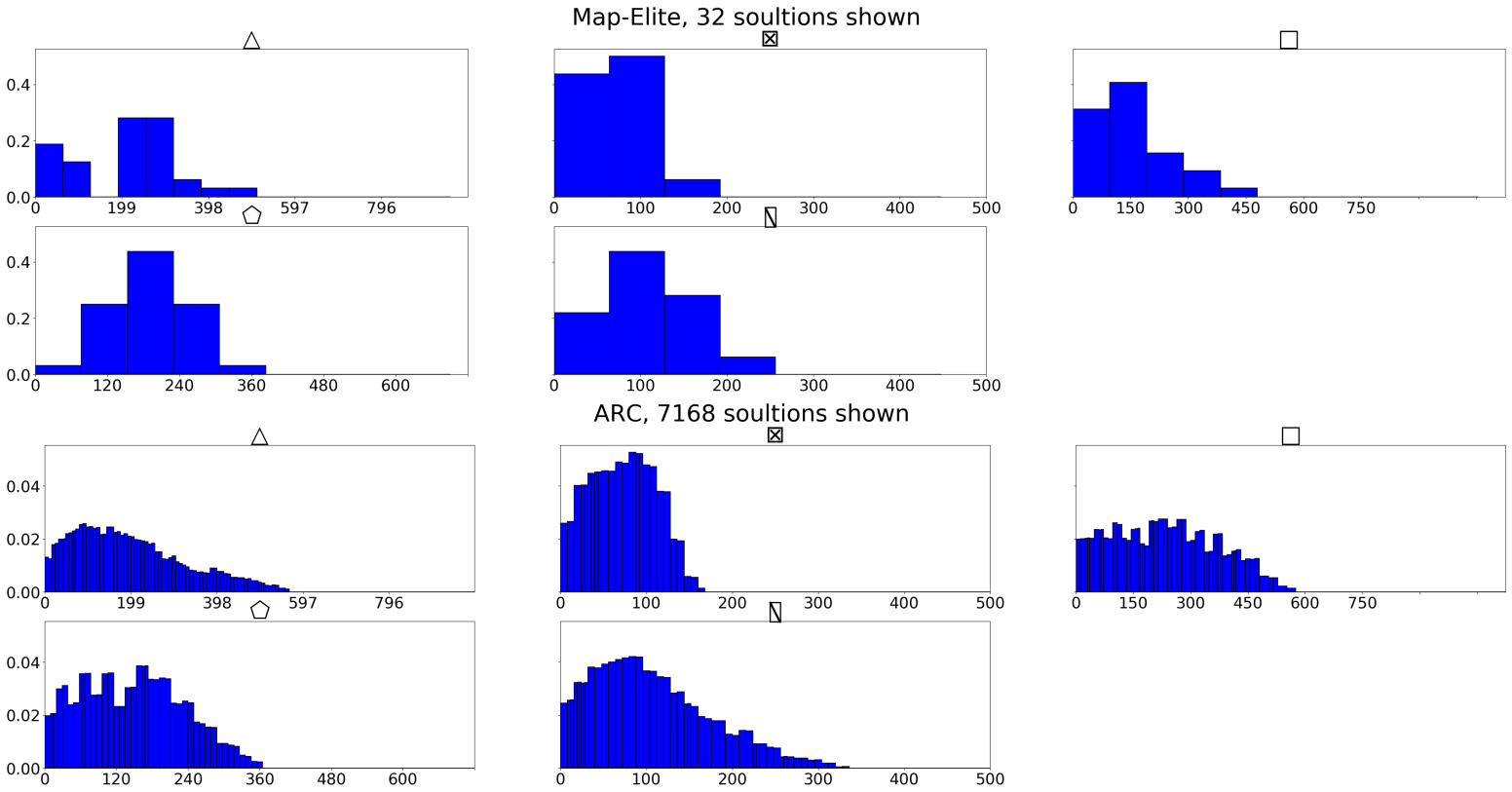


Figure 4.10: Number of overlapping filled cells across the subgraphs, as seen in Figure 4.1 (but there for a 2D space), used as the mapping dimensions in the ARC (bottom row) and MAP-Elites (top row) searches after 30,000 iterations for the 64 starting cell size condition.

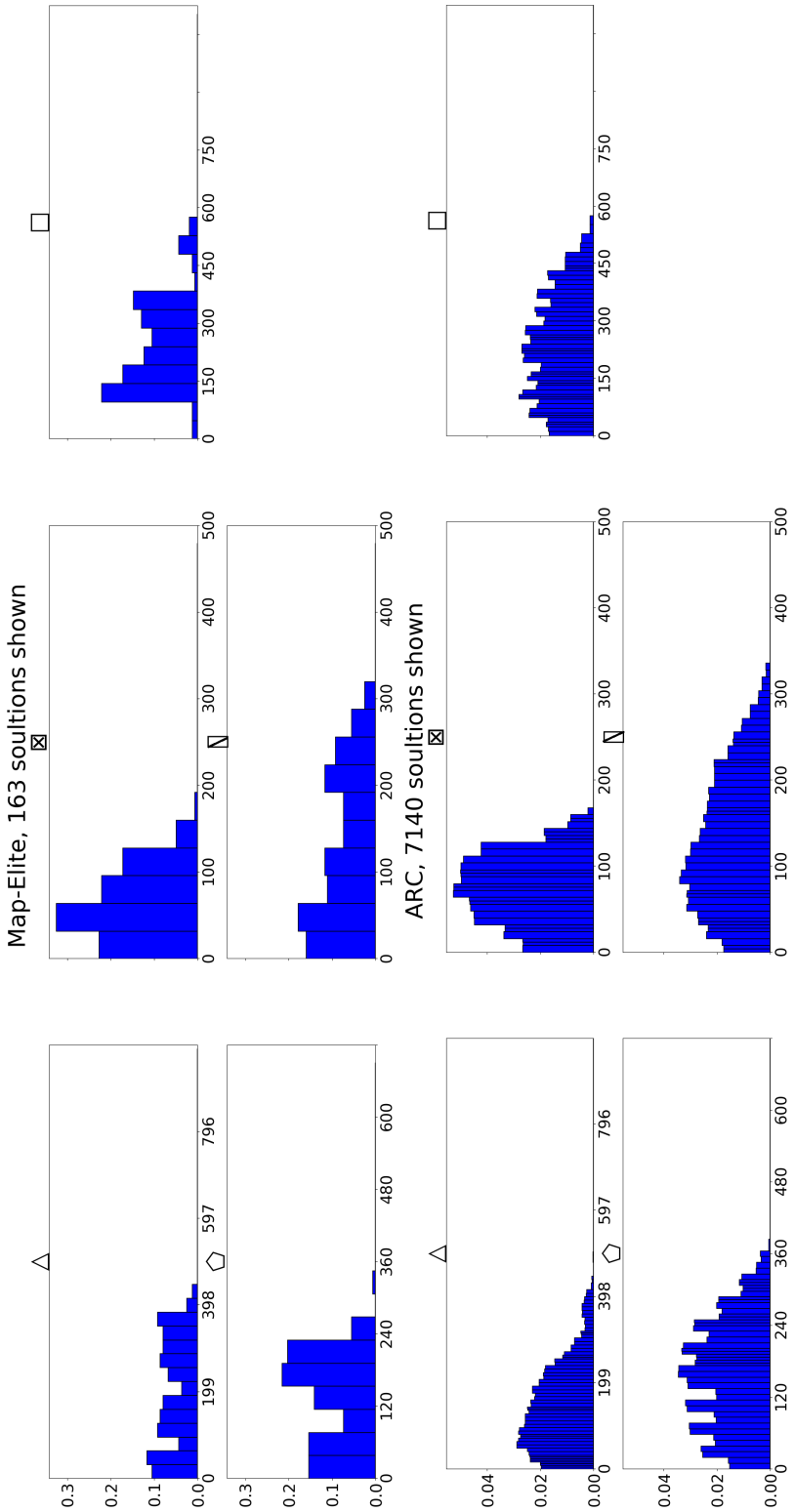


Figure 4.11: Number of overlapping filled cells across the five subgraphs, as seen in Figure 4.1 (but here for 2D), used as the mapping dimensions in the ARC (bottom row) and MAP-Elites (top row) searches after 30,000 iterations for the 32 starting cell size condition.

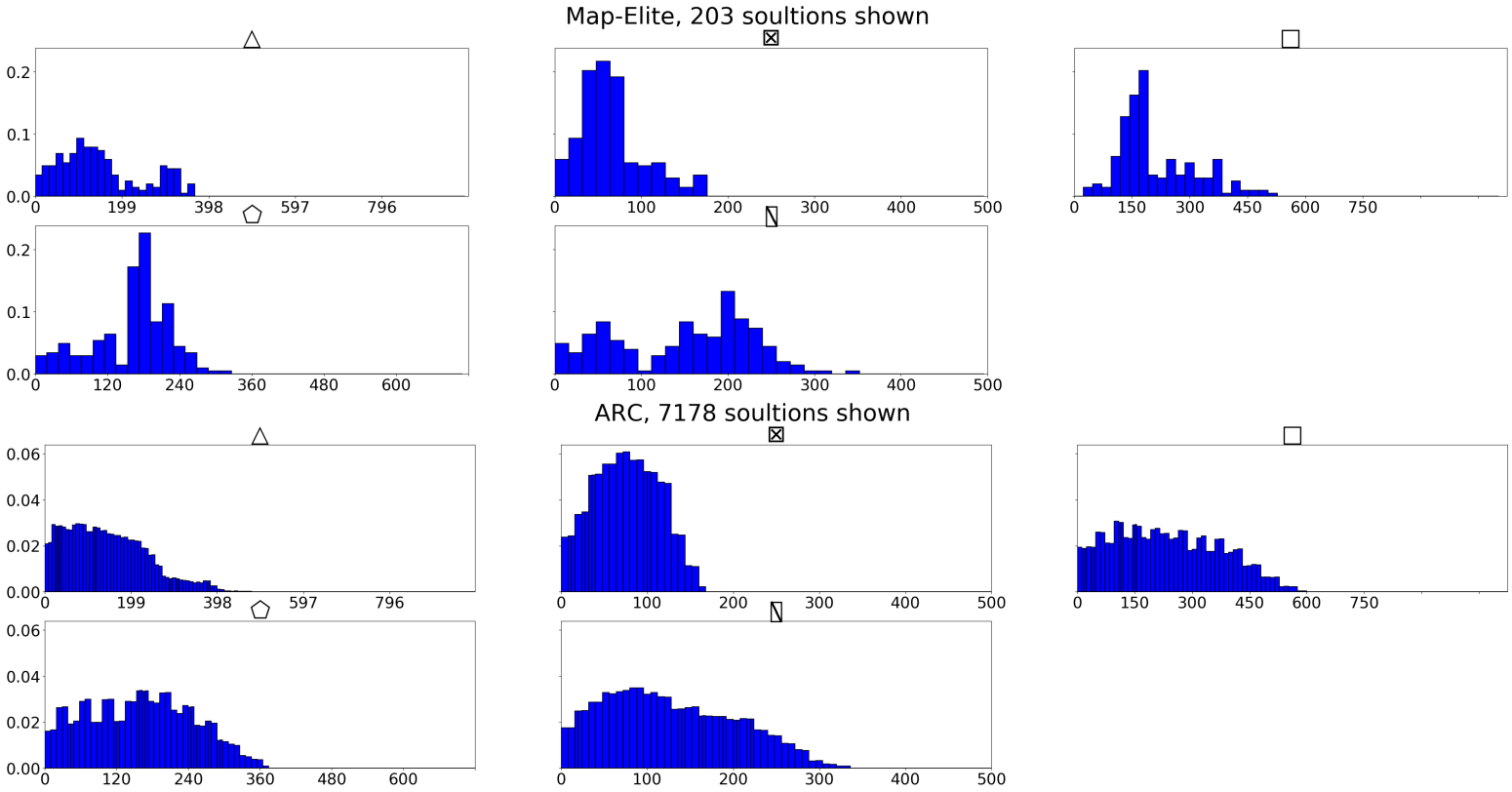


Figure 4.12: Number of overlapping filled cells across the five subgraphs, as seen in Figure 4.1 (but here for 2D), used as the mapping dimensions in the ARC (bottom row) and MAP-Elites (top row) searches after 30,000 iterations for the 16 starting cell size condition.

have high behavioural diversity is the research aim of quality diversity algorithms (QDs), and MAP-Elites-like algorithms are most commonly used for this purpose at present [78, 170, 79, 48]. These behaviour spaces can often promote movement to parts of the genome space that might otherwise be overlooked when diversity is maximised solely within the genome space. It is worth noting, this is very different to the past examples given in this paper so far (4.3.1 and 4.3.2, witch use a direct mapping of the genome space) and this of course make for a much more complex mapping of the space and the MoI chosen.

Here we demonstrate the effectiveness of the ARC method in a commonly used QD problem, by comparing it with MAP-Elites on the hexapod locomotion task [29] implemented using the Dynamic Animation and Robotics ToolKit (DART; [8]) in the mapping of fitness gradients across a behavioural feature space, not directly related to the genome. This task consists of the optimisation of the controller for a hexapod robot, as evaluated by the total forward distance covered in 5 s (i.e., the goal is to maximise the walking speed).

The controller has 36 parameters: three parameters for each of the two joints on the six legs of the robot (used as the genome here). These control an open-loop oscillator, which actuates each servomotor using a periodic signal (1 Hz). The three parameters for each of the joints are: (1) the amplitude of the oscillation; (2) its phase shift; and (3) its duty cycle (i.e., the fraction of each period during which the joint angle is positive).

The behavioural measure used here is the $DutyFactor$, which is defined as the proportion of time that each leg is in contact with the ground; thus, it has 6 dimensions, one per leg (see Figure 4.13). The equation used to calculate this value is given below (Equation 4.1), where $C_i(t)$ denotes the Boolean value of whether leg i is in contact with the ground at time t (i.e., 1: contact, 0: no contact). This is recorded at each time step (every 15 ms) and averaged over the total number of time steps T of the simulation, here 5 s (5,000 ms meaning 333

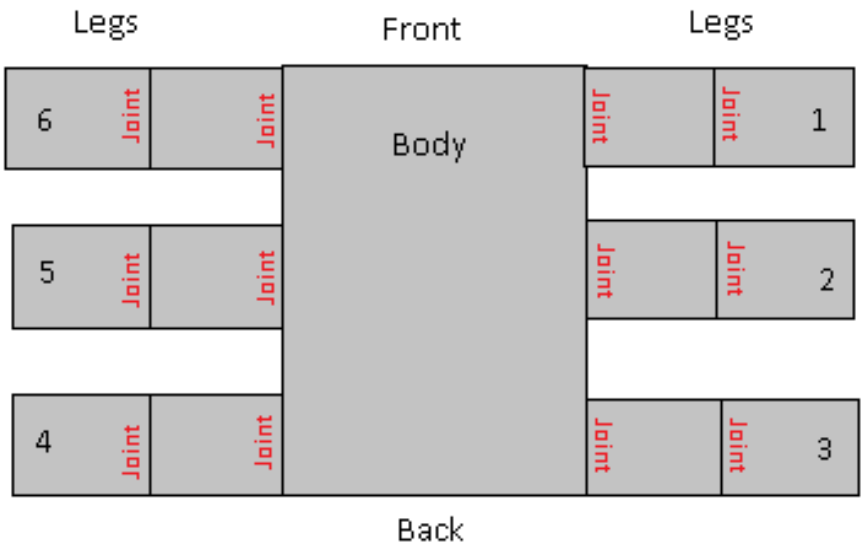


Figure 4.13: Position of each of the legs (labelled 1–6) and the joints of each leg for the hexapod robot in [29]

.

time steps).

$$DutyFactor = \left[\frac{\sum_{t=1}^{T} C_1(t)}{T}, ..., \frac{\sum_{t=1}^{T} C_6(t)}{T} \right] \in \mathbb{R}^6 \tag{4.1}$$

This problem landscape has been explored before in other works, such as [174] and [29], and there is agreement that there are 10,000 niches within the landscape for the particular problem of damage recovery. However, it is worth noting that this number of niches was obtained based on experimental results focusing on the differences in walking speed of the single best individual found after evolution (75,000 generations with 200 mutations per generation) from 20 independent runs, each of which used a different set number of niches (10, 100, 1,000, 10,000 and 100,000). Thus, although these results also looked at the diversifying effect of the conditions (varying number of niches) based on the performance of the population when different legs of the robot were damaged, this should not be mistaken for ground truth knowledge of the space. This is because if the performance of the

population is heavily reliant on the use of one of the legs, then it would be expected that performance should drop drastically in that damaged condition. This suggests diversity of the population if there is less of an effect. Thus, it would be fair to state that this space, at least as far as the relationship that each of the $DutyFactors$ has to the walking speed of the robot, is still relatively unknown. Here, we compare the ARC and MAP-Elites methods with three starting cell sizes: (1) 0.125 (262,144 possible starting cells); (2) 0.25 (4,096 possible starting cells); and (3) 0.5 (64 possible starting cells).

In each of the conditions (with five independent runs per condition), the algorithms were started with the same randomly generated starting population of 1,000 genomes and were run for 75,000 iterations (see Appendix 4.5.1 for list of the parameters used in each of the conditions). For the ARC conditions, the LL, NC, and Ro variables were set naively based solely on the guidelines given in Section 4.2.2 for the MoI used (i.e., variance in fitness) and the size of the problem landscape (LL=3, NC=2, and Ro=2). As such, they were set to favour exploration over exploitation of the space. The choices of using the variance in fitness as the MoI here meant that we focused on how each of these dimensions affected fitness, with the greatest attention paid to areas that have a more significant effect on walking speed. This is by far not the only choice of MoI that could be used and is poetically poorly sited for the optimisation of fitness, as discussed further in section 4.4. However, the proposed purpose of the ARC-MAP-Elite method is to focuses on the more complete mapping of the space with respect to the given question of interest, here the gradients of fitness to Duty Factor of the six legs, and thus in this purpose the purposed MoI fulfil this need. Further without focus on optimisation from the change events of ARC we can more confidently assert that any differences seen in performance be due to differences in the methods diversity of solutions over the methods diffidence in optimisation of cells.

First, examining the performance of the method at optimising the fitness, as shown in Figure 4.17, we observe the expected higher fitness in all of the MAP-Elites conditions compared

with the ARC conditions. Fitness decreases as the cell resolution increases above the 0.25 condition, supporting the assumption that 10,000 is the true number of niches in the fitness landscape (although only by an approximately 0.05 m/s difference in mean fitness). However, examining the distribution of filled cells of the MAP-Elites populations across each of the mapped $DutyFactors$ for the different starting cell sizes (examples are shown in the top panels of Figures 4.14, 4.15, and 4.16 for the 0.125, 0.25, and 0.5 starting cell size conditions respectively), we observe that much of this high performance comes at the cost of a lowered spread across the behaviour space, with significantly higher peaks in the distribution of solutions across all $DutyFactors$.

These higher peaks across the space of $DutyFactors$ suggest a focusing of MAP-Elites on some areas of the search space rather than others. Some areas of the space are not covered at all in the highest starting cell size conditions (in D2 between 0.5–1.0 and in D4 between 0.0–0.5, as seen in Figure 4.16). This low coverage could be attributable to the very low number of samples for this space (approximately 12 samples in 75,000 iterations). However, as we increase cell resolution, we still see a focusing of MAP-Elites on particular parts of the space that are not shared across the three MAP-Elites conditions or the ARC conditions. This is the case, for example, for starting cell size 0.25 (Figure 4.15) in D2 between 0.5–0.75, in D4 between 0.75–1.0, and in D5 between 0.0-0.25, and for starting cell size 0.125 (Figure 4.14) in D4 between 0.625–0.75. These differences in the locations of the peaks across each of the MAP-Elites examples is to be expected, since there is no selective pressure exerted on the direction of the search outside of each cell in the MAP-Elites method. Specifically. inside the cell there is an optimisation pressure towards the fittest solution, but if a mutation leads to an unfilled cell, there is no pressure placed on it movement; this is simply a movement to an unseen area of the space. This is not the same as the ARC method, in which the creation of new cells during the search via change events creates a selective pressure toward areas of highest cell density (i.e. areas where the largest

numbers of new cells have been created; that is, areas of highest MoI).

We examined the distribution of filled cells found by the ARC method during its search across each of the mapped $DutyFactors$ for the different starting cell sizes (examples shown in the bottom panels of Figures 4.14, 4.15, and 4.16 for the 0.125, 0.25 and 0.5 starting cell size conditions respectively). From the results, we can see that in all three conditions, the distributions of the filled cells are very similar for each of the $DutyFactors$. This suggests a convergence of factors independent of the starting cell size, with all of these distributions tending toward a higher frequency of cells with lower values for each duty factor (i.e., each of the legs spends less time on the ground) with a decreasing frequency of higher valued cells (i.e., each of the legs spends almost all of its time on the ground). This convergence suggests a difference in the density of higher fitness solutions across the space.

When the results from the ARC method are placed under the same starting cell size constraints as those of the MAP-Elites method (i.e., the ARC conditions run with a starting cell size of 0.25 will be put under the same storage restrictions as the 0.25 MAP-Elites conditions and so on for each of the starting cell sizes), we see that the ARC conditions were able to gain significantly greater coverage scores compared with the MAP-Elites conditions for all of the starting cell sizes (see panels b and d of Figure 4.18). Furthermore, when all of the ARC conditions are compared using the largest cell size restrictions (0.5), the results show that they were not only able to gain significantly greater coverage scores (suggesting that the ARC solutions are spread much more widely across the space of solutions), but also all converged to a very similar level of around 0.6 after 75,000 generations (see panel a of Figure 4.18).

These results suggest that, by focusing the search on the areas of highest MoI (here the variance in fitness) by decreasing cell sizes in these areas (thereby increasing the representation of the area in the population and proportionally decreasing the mutation size), the ARC method is able to identify areas of the space that are more likely to lead to

unexplored regions of the space.

This is observed despite the normal association between a decrease in average mutation size resulting in decreasing exploration. Additionally, the MoI used here does not provide any direct reward for this increasing coverage; that is, there is no direct optimisation of the number of new cells found.

The importance of this added diversity is revealed by the fact that the ARC method shows a significantly smaller decrease in performance from the non-damaged to the damaged condition (removal of leg 2) compared with MAP-Elites, as shown by Figure 4.19. This reduction in the effect of damage on the ARC solutions is seen across all starting cell sizes. However, this reduction does increase with increasing starting cell size (i.e., the difference is the lowest in the 0.5 ARC condition). This suggests that the extra cells filled by the ARC method contribute solutions to the population by providing significantly different movement patterns. These demonstrate better resistance to unforeseen damage to the robot, such as the loss of a leg, as tested here. This resilience to damage is even present when the ARC population is converted to match the fixed cell size of MAP-Elites (in the same way as for the coverage measures discussed previously), suggesting that this effect is not just due to the greater number of samples stored by the ARC method.

In summary, these results show that despite its low performance, the ARC method is able to provide a significantly greater coverage of possible solutions, even without knowledge of the appropriate staring cell size. Even when it is started at the lowest starting cell size, ARC is able to provide a greater range of behavioural solutions, as demonstrated by the increased resilience to unforeseen damage. The lower performance of this method could be addressed by re-balancing the LL, Ro, and/or NN values to increase the number of revisits required before a change event takes place, thus increasing the amount of optimisation performed within each cell. However, it should be highlighted again that is was not the focus of this work and as such, the MoI used here (variation in fitness within a cell) is not well suited to

focusing the search towards high performance. Rather, it focuses it towards the areas of greatest influence on the performance, positively or negatively.

4.4 Discussion

We have presented a new algorithm, the ARC MAP-Elites method, for automatically tailoring cell sizes to reflect the levels of interest across the mapped space in the MAP-Elites method. We detailed the factors that should be considered in deciding on an appropriate MoI, and further demonstrated the sensitivity of the method to the three controlling variables, which can be used to tailor the level of exploitation vs exploration of the search, as well as how these variables are affected by the choice of MoI used. Three example problem landscapes were explored and benchmarked against the original MAP-Elites method. In each of these very different problem spaces, the ARC method was shown to outperform the MAP-Elites method in the identification and representation of the areas of interest within the space when it was started with the larger cell size. This can be seen by the number of global optima found in the niche examples (Figure 4.8) and in the coverage scores in the network and hexapod scenarios (Figures 4.9 and 4.18, respectively). This is to be expected as the larger the cell size the ARC method is started with, the greater the freedom it has to tailor the size of cells across the space. Indeed, the ARC method never increases the size of a cell, it only ever decreases it. This makes the starting cell size a hard limit to the maximum possible size of cells across the space. Nevertheless, even in the smaller starting cell size conditions, the ARC method showed a grater range of solutions, as evidenced by the distribution of filled cells in the network examples (Figures 4.10, 4.11, and 4.12) and by the resilience to unforeseen damage shown by the solutions in the hexapod scenario (top plane of Figure 4.19).

Furthermore, in all three examples shown here, the results of the ARC method showed convergence to very similar distributions both for the number of filled cells and for the cell

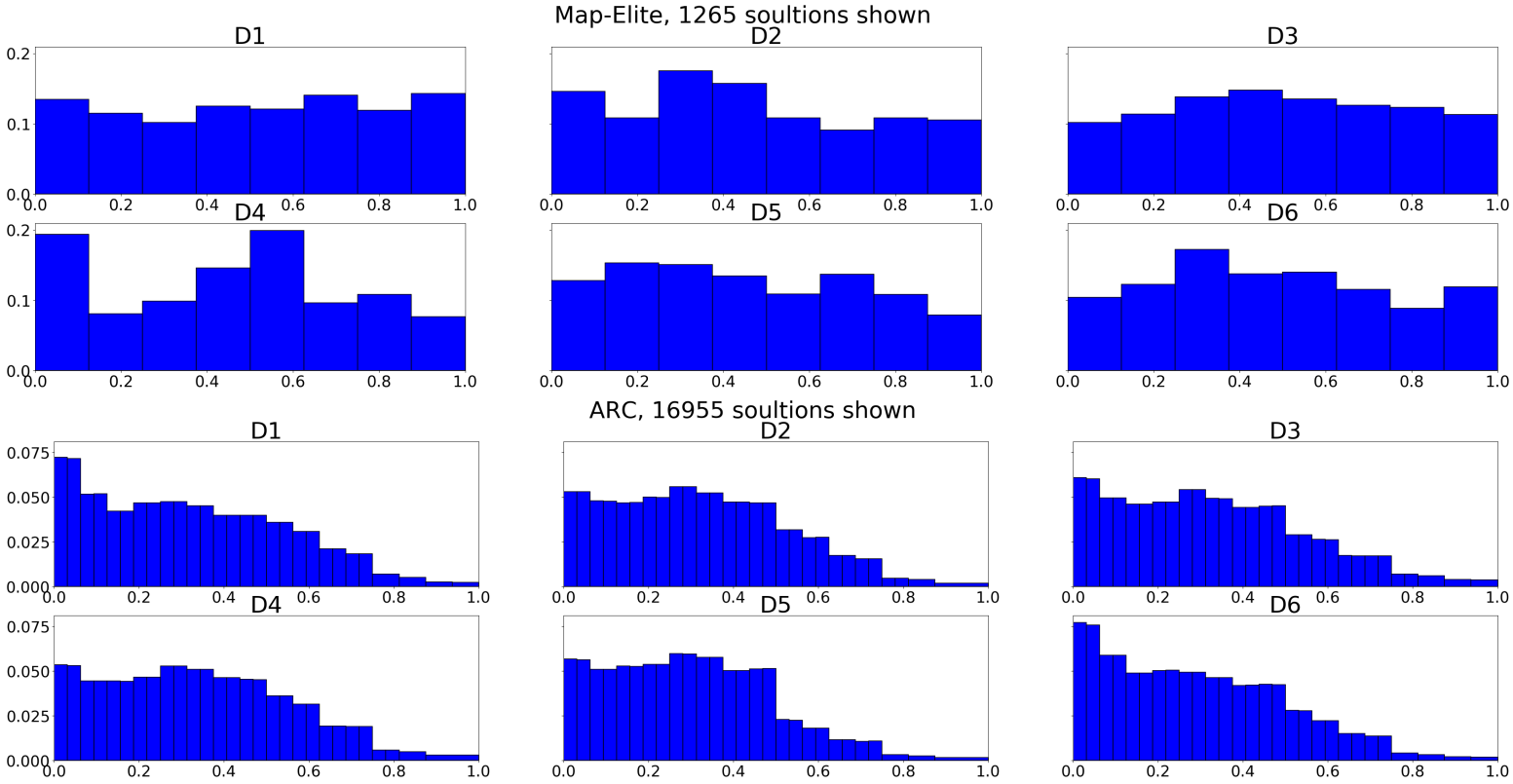


Figure 4.14: Number of overlapping filled cells (as shown in Figure 4.1) after 75,000 iterations for the 0.125 starting cell size condition, with the six $DutyFactors$ ($D1 = leg1$, $D2 = leg2$, and so on; see Figure 4.13) used as mapping dimensions in the ARC (bottom row) and MAP-Elites (top row) methods.

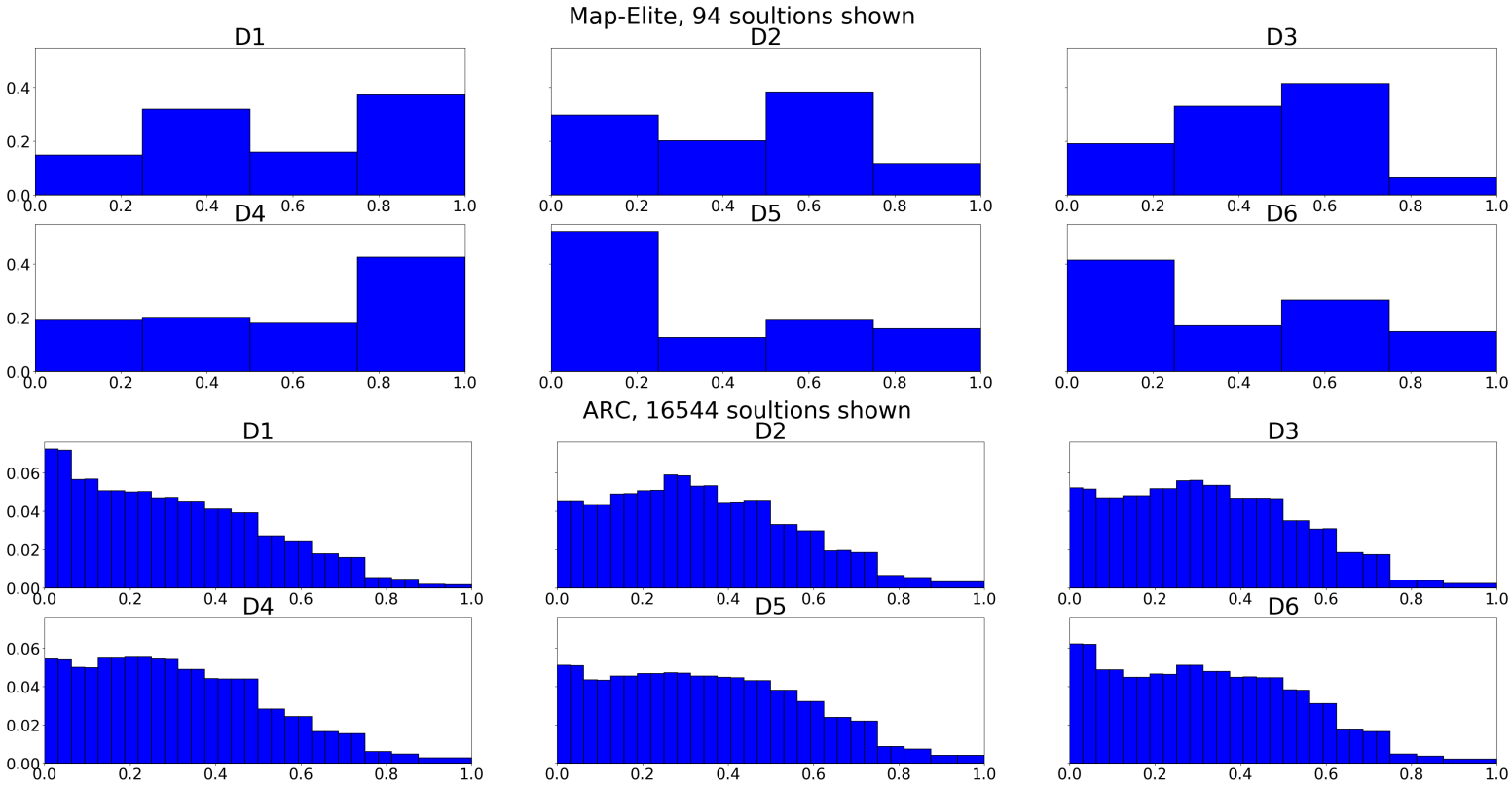


Figure 4.15: Number of overlapping filled cells (as shown in Figure 4.1) after 75,000 iterations for the 0.25 starting cell size condition, with the six $DutyFactors$ ($D1 = leg1$, $D2 = leg2$, and so on; see Figure 4.13) used as mapping dimensions in the ARC (bottom row) and MAP-Elites (top row) methods.

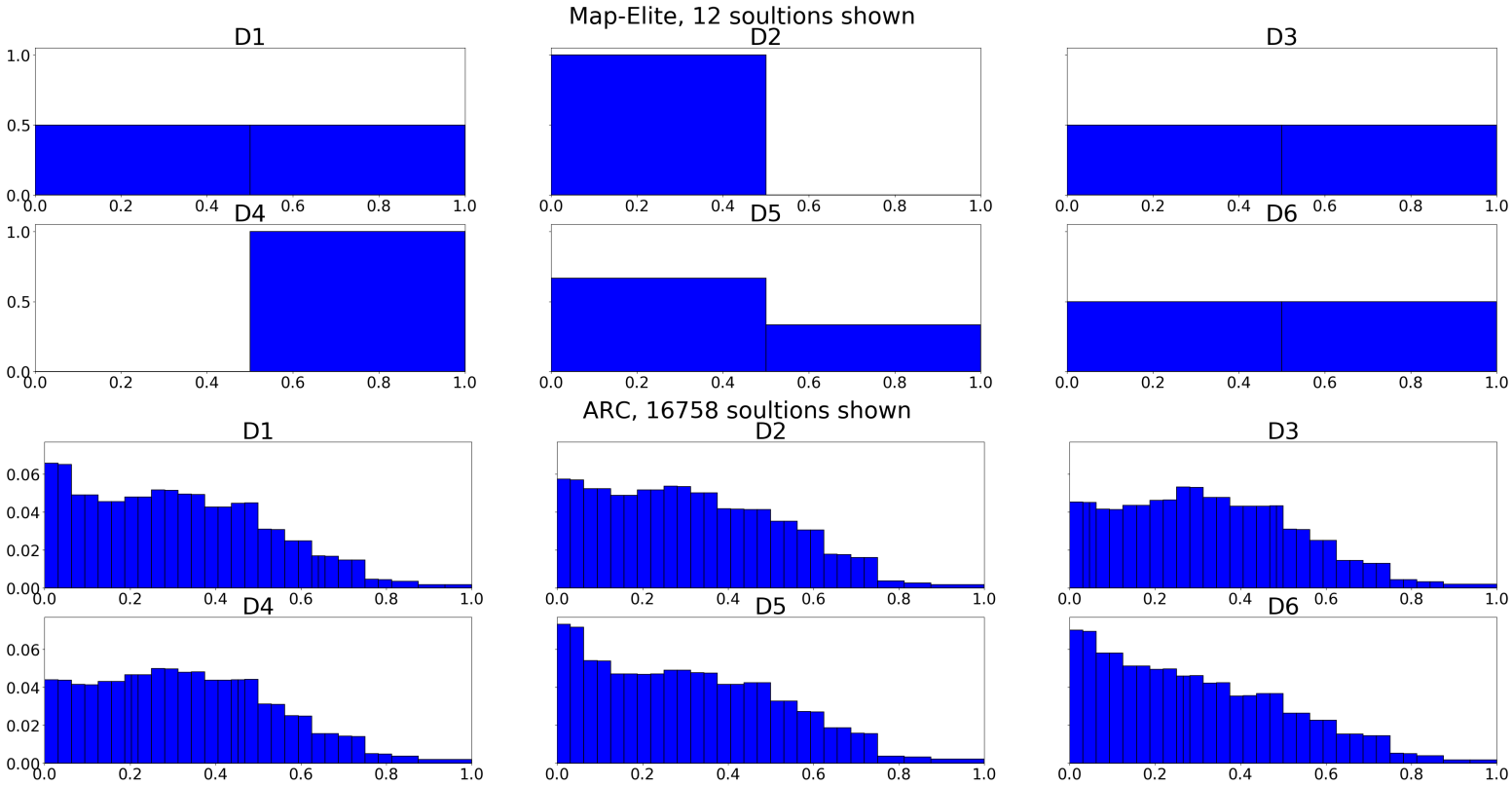


Figure 4.16: Number of overlapping filled cells (as shown in Figure 4.1) after 75,000 iterations for the 0.5 starting cell size condition, with the six $DutyFactors$ ($D1 = leg1$, $D2 = leg2$, and so on; see Figure 4.13) used as mapping dimensions in the ARC (bottom row) and MAP-Elites (top row) methods.

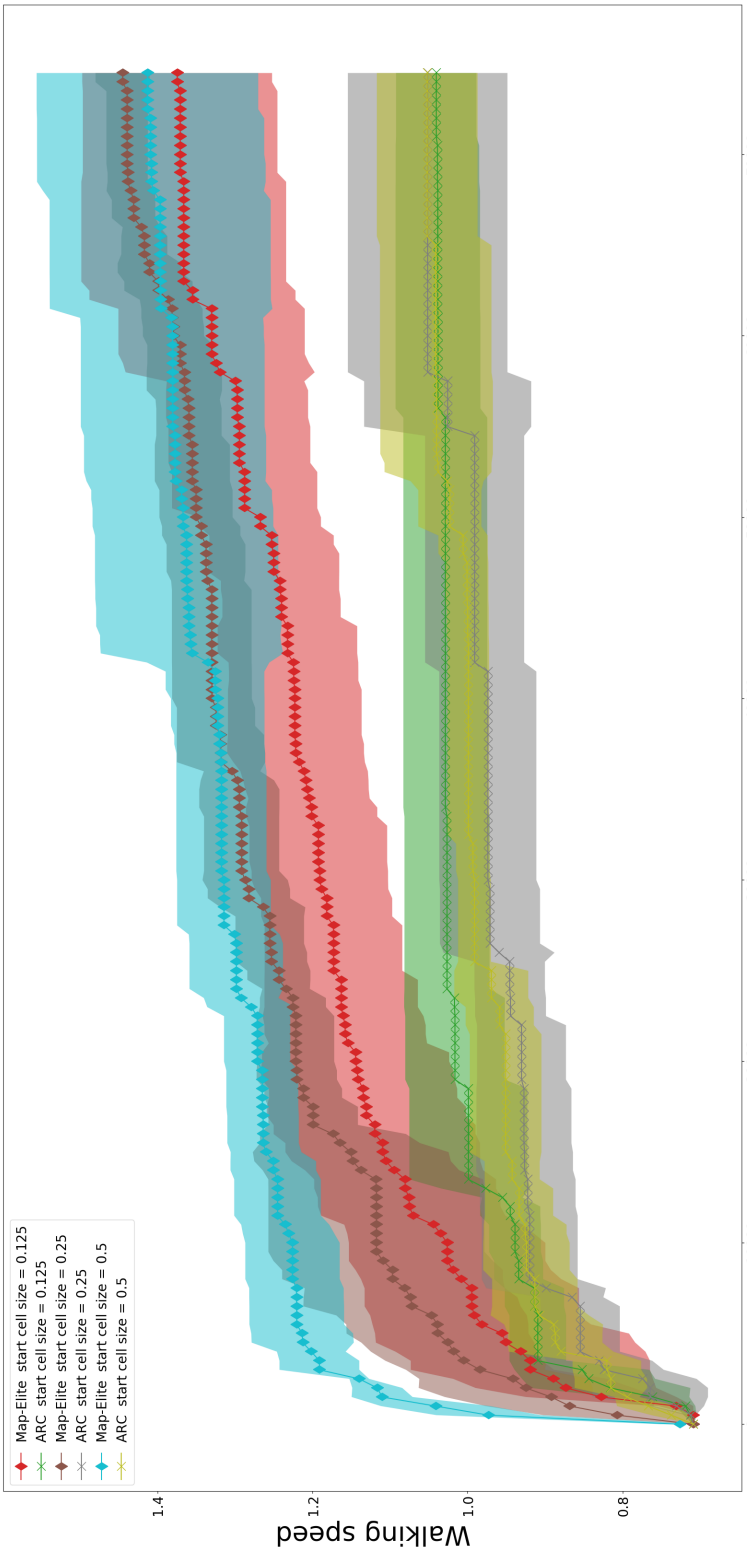


Figure 4.17: Best performance from the population after 75,000 iterations over five independent runs of each condition.

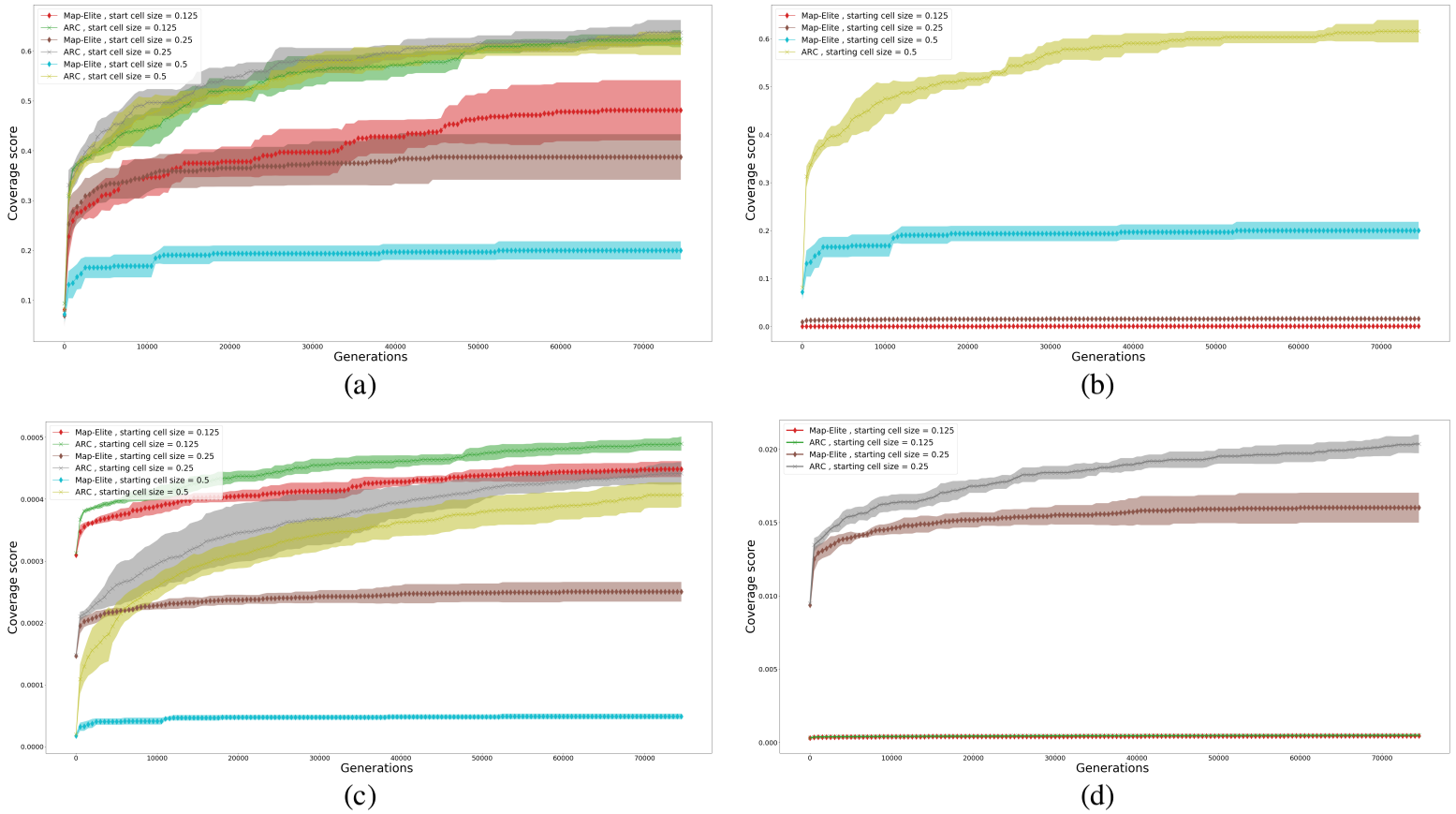


Figure 4.18: Coverage (i.e., the total number of cells found divided by the total possible number of cells for that starting cell size) of the population after 75,000 iterations over five independent runs of each condition. Panel (a) shows the coverage when all conditions were compared using the largest cell size (0.5). Panel (c) shows the coverage when all conditions were compared at the smaller cell size (0.125). Panels (b) and (d) (the right-most column of panels) show the coverage when the cell size of the coverage was matched to the starting cell sizes of the conditions shown.

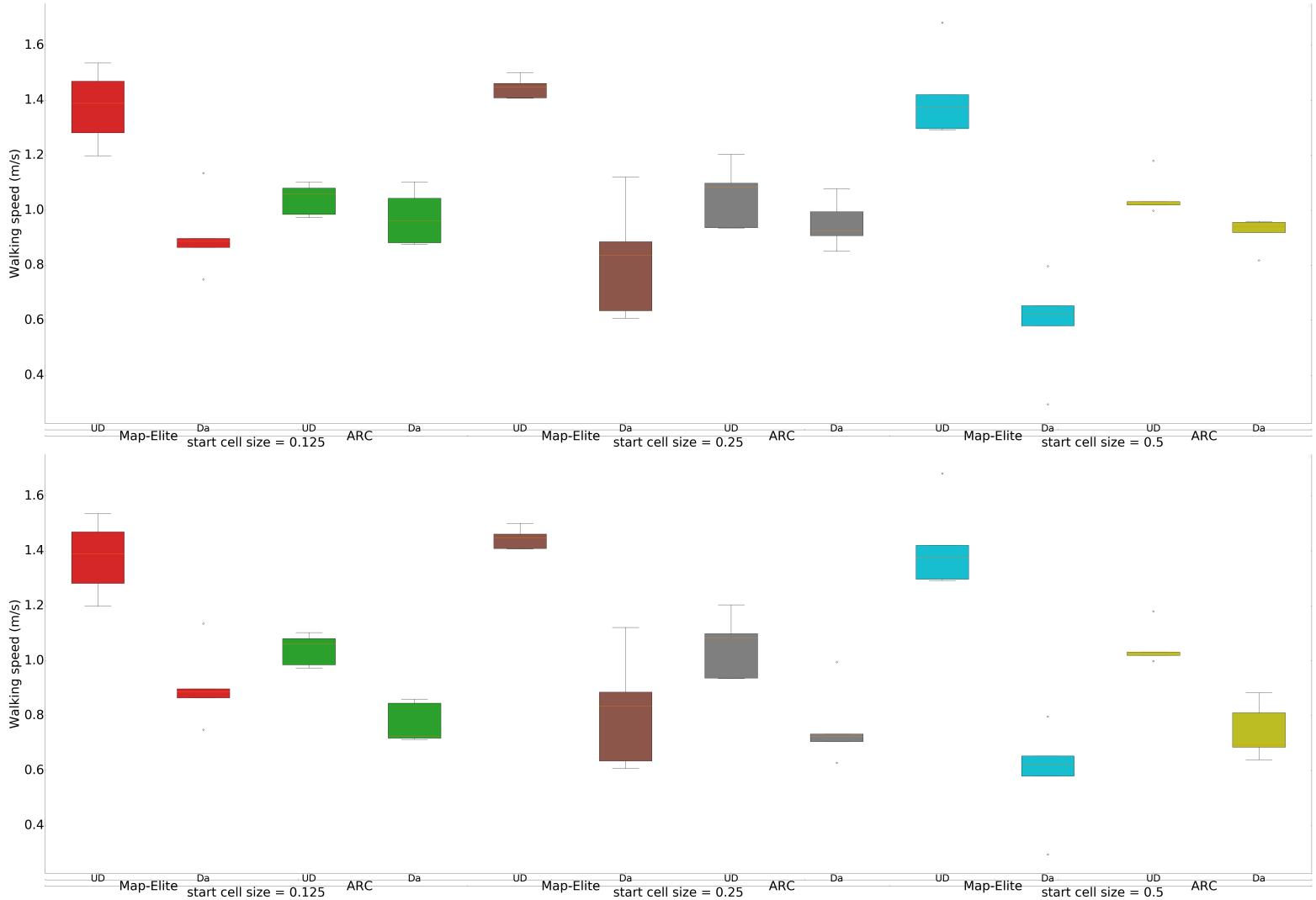


Figure 4.19: Shows the best performance from the population after 75,000 iterations over five independent runs of each condition, for the undamaged (i.e., measure used to evaluate fitness during evolution; left-most plot for each of the pairs, labelled "UD") and damaged (i.e., with leg 2 removed; right-most plot for each of the pairs, labelled "Da") robot simulations. The top panel shows results using the original starting cell sizes for each condition.

sizes across each of the mapped dimensions. This suggests that the results found by the ARC method are much more able to deal with an unknown mapped space. In such a space, in which the appropriate cell size is unknown, MAP-Elites showed no such convergence across the different starting cell sizes. This convergence was also seen in the levels of performance of the ARC method, regardless of starting cell sizes, across all problem landscapes in which a measure of fitness was used. This further supports the conclusion that the ARC method is robust to starting cell sizes.

It is worth noting that the ARC method showed a substantially lower level of performance than MAP-Elites. This is to be expected as the focus of our work is on the exploration of the problem spaces, rather than on optimisation towards the highest fitness solutions within these spaces. Nevertheless, there are many modifications that could be made to the ARC method to improve the level of optimisation within these landscapes, possibly even without significantly decreasing the coverage found here. First of all, the variables controlling the change events (LL, Ro, and NN) could be altered to give a greater focus on the exploitation of larger cells by increasing the level of revisiting required before allowing change events (i.e., by increasing LL and/slash or Ro). This would likely change the balance of exploration, as it will cause change events to occur less often (see Section 4.2.2), but this effect could be lessened by increasing the number of cells changed per change event (NN). This, however, this would only be advisable in problem landscapes in which the MoI is easily assessed (i.e., a high confidence can be given to the accuracy of the MoI after obtaining just a few samples from each cell). Of course, as discussed in Section 4.2.1 and 4.2.2, the exact effect of these changes will depend on the problem space and on the end goal of the search employed. Alternatively, many methods designed to improve the optimisation ability of the original MAP-Elites method could be employed here without the need for significant changes to the ARC method. For example, the CMA-ME (Covariance Matrix Adaptation MAP-Elite) method [50] would require no changes to the ARC method, but

merely the addition and updating of emitters to match the changing cell sizes. This could significantly improve the optimisation of the ARC method, while adding to the diversity of solutions gained from the CMA-ME method. This results from the fact that the main gain of diversity in the CMA-ME method derives from its use of the pre-sized grid of the MAP-Elites method, which ARC has been shown to outperform.

However, we stress that the aim of the ARC MAP-Elites method was not the optimisation of fitness. Rather, it was to provide a richer level of detail on the relationship between each of the mapped dimensions and their level of performance and/slash or the MoI employed, if a fitness-based MoI was not used. This is highlighted particularly in the hexapod example above, in which the converged distribution of filled cells across the $DutyFactors$ for each of the legs of the robot gives insight into how the time each leg spends on the ground affects the speed of the robot. For example, although for the majority of the legs there seems to be a decreasing number of high-performing solutions with an increasing $DutyFactor$, for the middle legs (legs 2 and 5; see Figure 4.13), there appears to be a peak closer to 0.3. This suggests that, although a short time spent in contact with the ground may mostly affect the speed of the outer legs, the middle legs require a greater degree of contact with the ground (perhaps to provide stability to the robot). Here the MoI used (variability in fitness) allowed the relationship between the $DutyFactors$ and fitness to be highlighted. However, this MoI is unsuited for optimising fitness, because it focuses on the areas of greatest change in fitness with respect to the mapped dimensions. This is thus unlikely to result in a focusing of the search on the areas of highest fitness unless the spaces are known for highly variable fitness landscapes.

Although we only highlight two MoIs in this paper, there are many different MoI that could be chosen to investigate alternative questions about the space. Such MoIs should ideally still be based on the added information provided by revisiting seen areas, but this does leave many interesting possibilities (i.e., it should allow for a natural decrease in effect with

decreasing cell size, as discussed in Section 4.2.1). For example, in spaces in which we expect to see differences in the distribution of realisable solutions (i.e., in which we are interested in highlighting the rarest solutions, such as the one-legged walking robot scenario used here), one could base the MoI on the lowest number of revisits. This would likely have a similar effect to the use of the SHINE method [157]; that is, the search would focus on the areas of the space with the lowest density of solutions. However, it would might benefit from the localisation of the search around these "rare" solutions resulting from the decreasing cell size and the linked mutation size of the ARC method. Another interesting MoI could be based on some version of the Curiosity Score [30] of each cell, which is the propensity of each cell to generate offspring that are added to the collection, with resetting after each change event. This would likely focus the search towards different areas of the space as the search goes on, with the level of curiosity being dynamically linked to the parts of the space that have already been covered. These would mean that parts of the space with high curiosity will likely be rapidly explored leading to a rapid decrease in the Curiosity Scores of the same areas. As such, this particular MoI may be best suited to rapidly exploring a space, perhaps in which there is a limited number of mutations that can be made during the search, or paired with some other method of increasing the cell size with a decreasing MoI (Curiosity Score). Either way, it would be an interesting way to explore the relationship of "stepping stones" (i.e., intermediate sub-optimal solutions that aid in the mitigation of premature convergence to local optima [29, 93, 141]) to the given mapped dimensions.
Of course, the ARC method still suffers from the same limitations as MAP-Elites, particularly in that it assumes that full coverage of the space is possible in a reasonable number of iterations of the search; that is, that the number of potential cells is so high that there will never be enough revisits to gain an accurate assessment of the space for change events to occur. In a high-dimensional mapping space, this is in fact highly likely, even when the method is started with a very small number of starting cells (i.e., a very coarse

resolution).

In such high-dimensional problem spaces, it should be possible to circumvent this issue by combining ARC with alternative methods of defining cells, which are used to solve these issues with MAP-Elites (such as the CVT methods of [174].

This kind of dimensional scaling of the ARC method would require further work in order to show its effectiveness. However, it should be possible to change the CVT "cells" such that the space covered by a selected cell can be shared evenly across split cells after a change event. Note that this would require a CVT-like division of the space within the selected cell, rather than just increasing the number of niches used by the CVT method; this is because the CVT method assumes an even distribution of niches across the space. How this kind of scaling would affect the ability of our method to characterise the relationship between each of the mapped dimensions and the MoI would of course have to be investigated. However, this could potentially provide a new way to minimise the effect of adding cells in higher dimensions, thus increasing the effectiveness of the method for optimising fitness.

4.5 Appendix

4.5.1 Appendix A: Experimental parameters and setups used

This appendix lists the parameters and setups used in the above experiments, namely niches (Section 4.3.1), network exploration (Section 4.3.2) and the hexapod walking task (Section 4.3.3). All experiments were assessed using five independent runs for each of the conditions and were initialised using the same starting population for each condition, irrespective of the method used (i.e., ARC and MAP-Elites runs used the same starting population for each run in each condition). Individual parameters used for the different experiments are listed in Table 4.5.1, below.

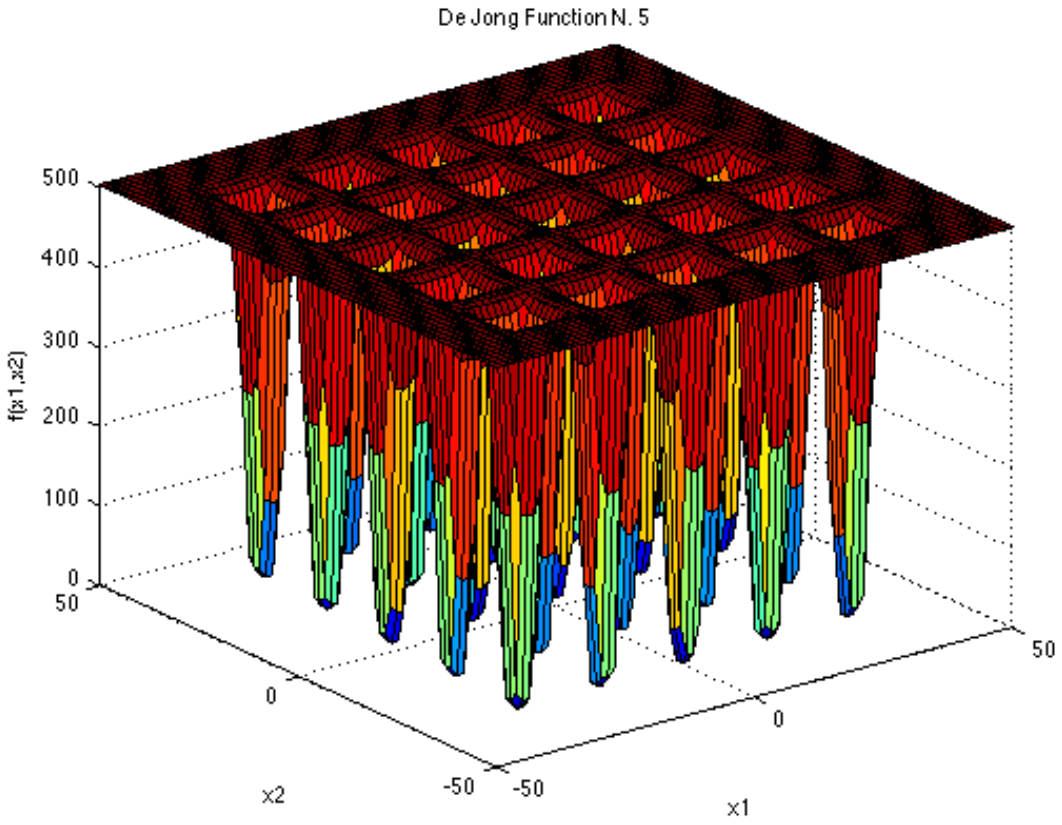
Parameters of Evolutionary Algorithms						
Experimental condition	*NN*	*LL*	*Ro*	**Starting cell sizes**	**Chance of mutation per gene**	**Mutation range**
Sensitivity tests	n/a	n/a	n/a	0.25, meaning 16 cells at the start	100%	[cellsize to cellsize*2] i.e., "out of cell" condition
Mutation range tests	2	3	2	0.25, meaning 16 cells at the start	100%	n/a
Niche landscape functions	2	3	2	0.5 and 0.25, meaning 4 and 16 cells respectively at the start	100%	50% [cellsize to cellsize*2] and 50% [cellsize /2 to cellsize] i.e., the 50% condition
Hexapod robot controls	2	3	2	0.5, 0.25, and 0.125, meaning 64, 2144, and 4096 cells respectively at the start	30%	50% [cellsize to cellsize*2] and 50% [cellsize /2 to cellsize] i.e., the 50% condition
Network structure exploration	2	3	2	64, 32, and 16, meaning 104,669, 3,349,408 and 107,181,072 cells respectively at the start	Diophantine mutation (see Section 4.3.2)	[cellsize to cellsize*2] i.e., "out of cell" condition

Table 4.1: Experimental parameters

4.5.2 Appendix B: Artificial 2D optimisation landscapes

This appendix details the three artificial 2D optimisation landscapes used in Section 4.2 for demonstrating the sensitivity of different problem landscapes to the control variables (LL, Ro, and NN) . Note that these landscapes were chosen before becoming aware of the work of [20], which compares the reliability of grid-based quality-diversity algorithms using artificial landscapes. Our selection does not include the Rastrigin function landscape, although it was tested in preliminary experiments. All fitness results from these landscapes (here shown as $f(\mathbf{x})$ in the equations) and coordinates were normalised to a range of 0 to 1 after calculating the equations below.

DE JONG FUNCTION N. 5

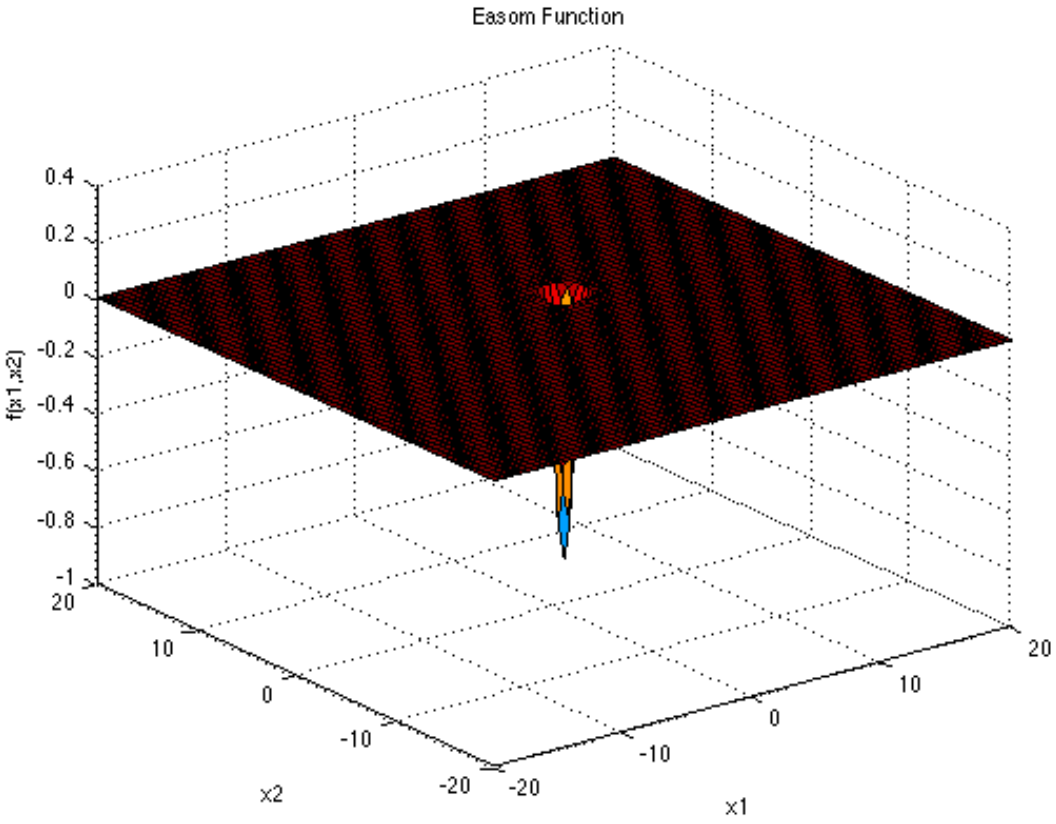


$$f(\mathbf{x}) = \left(0.002 + \sum_{i=1}^{25} \frac{1}{i + \left(x_1 - a_{1i}\right)^6 + \left(x_2 - a_{2i}\right)^6} \right)^{-1}, \text{ where}$$

$$\mathbf{a} = \begin{pmatrix} -32 & -16 & 0 & 16 & 32 & -32 & \dots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & \dots & 32 & 32 & 32 \end{pmatrix}$$

Figure 4.20: DE JONG FUNCTION N. 5, figure and equation taken from https://www.sfu.ca/~ssurjano/dejong5.html (last accessed on 5/3/2020).

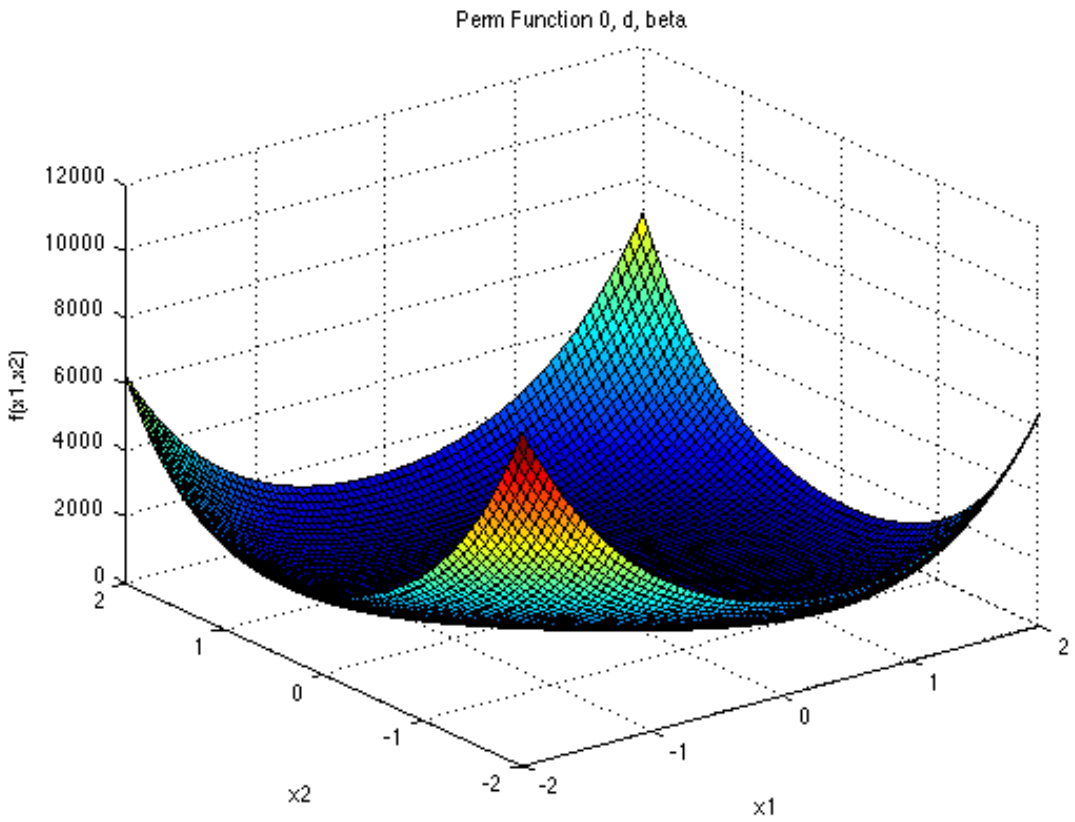
EASOM FUNCTION



$$f(\mathrm{X}) = -cos(\mathrm{x}_1)cos(\mathrm{x}_2)exp(-(\mathrm{x}_1 - \pi)^2 - (\mathrm{x}_2 - \pi)^2)$$

Figure 4.21: EASOM FUNCTION, figure and equation taken from https://www.sfu.ca/~ssurjano/easom.html (last accessed on 5/3/2020).

PERM FUNCTION 0, D, BETA



$$f(\mathbf{x}) = \sum_{i=1}^{d} \left(\sum_{j=1}^{d} (j+\beta) \left(x_j^i - \frac{1}{j^i} \right) \right)^2$$

Figure 4.22: PERM FUNCTION 0, D, BETA, figure and equation taken from https://www.sfu.ca/~ssurjano/perm0db.html (last accessed on 5/3/2020).

4.5.3 Appendix C: Composition function

All functions below were obtained and implemented from the code available at https://github.com/mikeagn/CEC2013 (last accessed on 5/3/2020) and based on the function laid out in [95]. Being composition functions, the functions laid out below make use of two or more established functions in order to create the end landscape. These are detailed in Equations 4.5.3, 4.3, 4.4, 4.5, and 4.6 for the composition function used here. Full information regarding their construction can be obtained from [95].

$$f_S(\vec{x}) = \sum_{i=1}^{D} x_i^2 \tag{4.2}$$

Sphere function

$$f_G(\vec{x}) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \tag{4.3}$$

Grienwank function

$$f_R(\vec{x}) = \sum_{i=1}^{D} \left(x_i^2 - 10\cos\left(2\pi x_i\right) + 10\right) \tag{4.4}$$

Rastrigin function

$$\begin{aligned} f_W(\vec{x}) = &\sum_{i=1}^{D} \left(\sum_{k=0}^{\text{kmax}} \alpha^k \cos\left(2\pi\beta^k \left(x_i + 0.5\right)\right) \right) \\ &- D \sum_{k=0}^{\text{kmax}} \alpha^k \cos\left(2\pi\beta^k (0.5)\right) \end{aligned} \tag{4.5}$$

Weierstrass function

$$\begin{aligned} F8(\vec{x}) &= \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \\ F2(x) &= \sum_{i=1}^{D-1} \left(100\left(x_i^2 - x_{i+1}\right)^2 + \left(x_i - 1\right)^2\right) \\ E \cdot 8F2(x) &= F8F2\left(x_1, x_2, \ldots, x_D\right) \\ &= F8\left(F2\left(x_1, x_2\right)\right) + F8\left(F2\left(x_2, x_3\right)\right) + \ldots \\ &\quad + F8\left(F2\left(x_{D-1}, x_D\right)\right) + F8\left(F2\left(x_D, x_1\right)\right) \end{aligned} \tag{4.6}$$

Expanded Griewank plus Rosenbrock function (EF8F2)

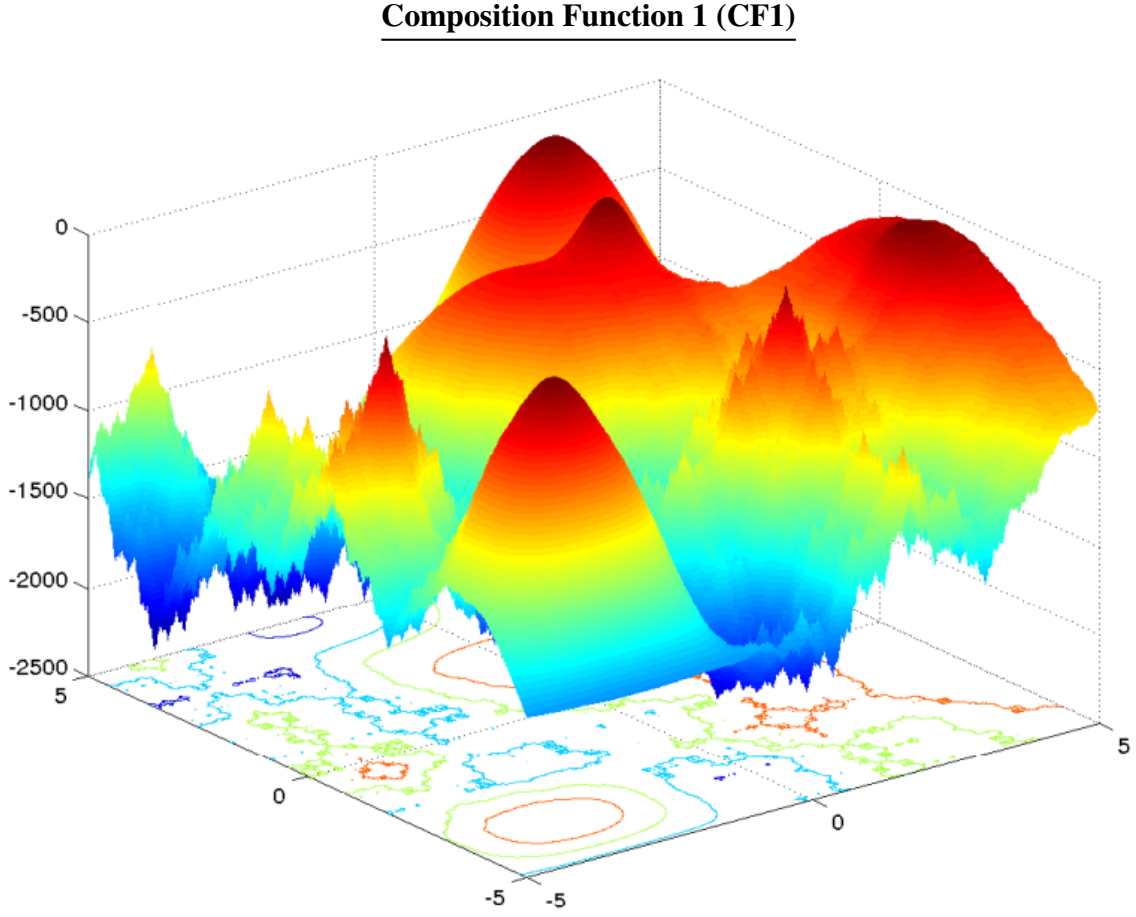


Figure 4.23: Composition Function 1 with fitness on the vertical axis. CF1 is constructed based on six basic functions (2 × Grienwank function, 2 × Weierstrass function, and 2 × Sphere function), yielding a total of six global optima in the range of $A_D = [5; 5]^D$, here normalised to 0–1. Figure taken from [95].

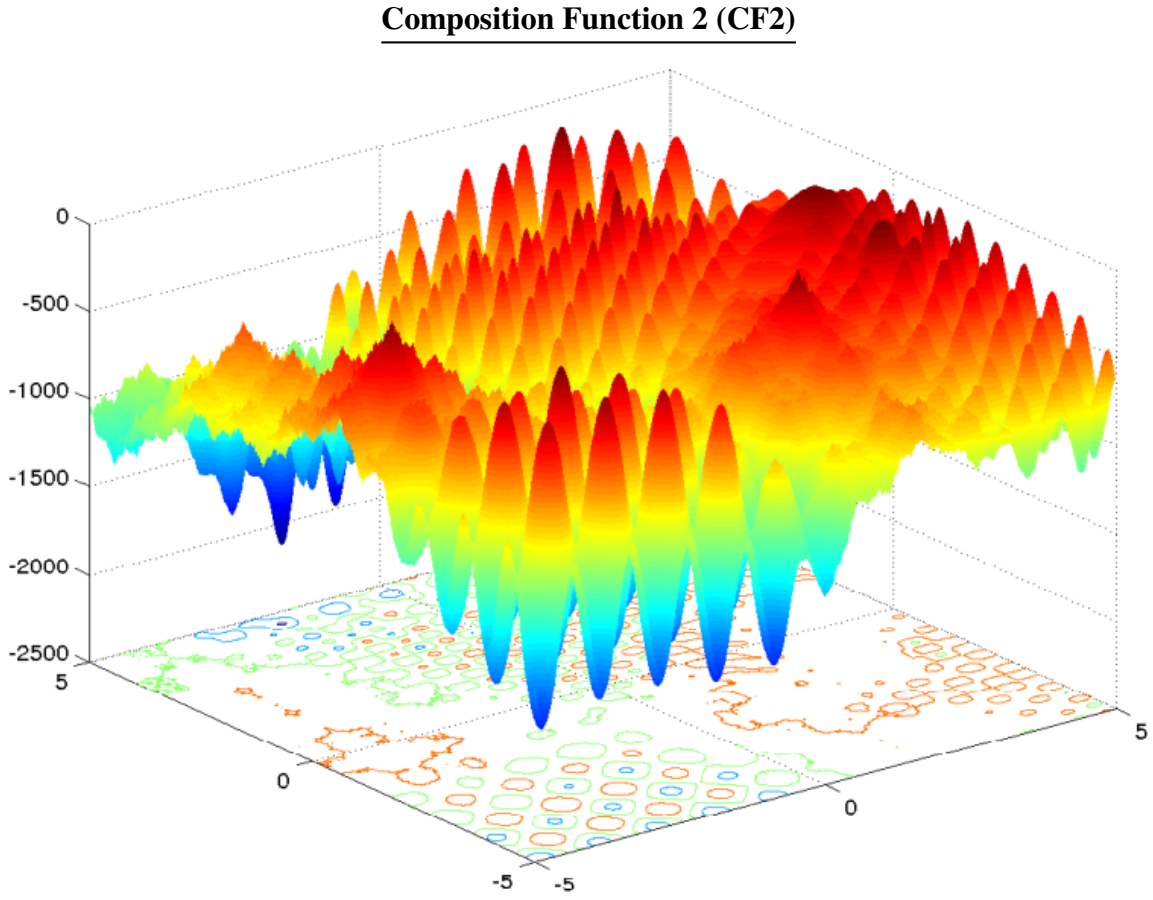


Figure 4.24: Composition Function 2 with fitness on the vertical axis. CF2 is constructed based on eight basic functions (2 × Rastrigin function, 2 × Weierstrass function, 2 × Griewank function, and 2 × Sphere function), yielding a total of eight global optima in the range of $A_D = [5; 5]^D$, here normalised to 0–1. Figure taken from [95].

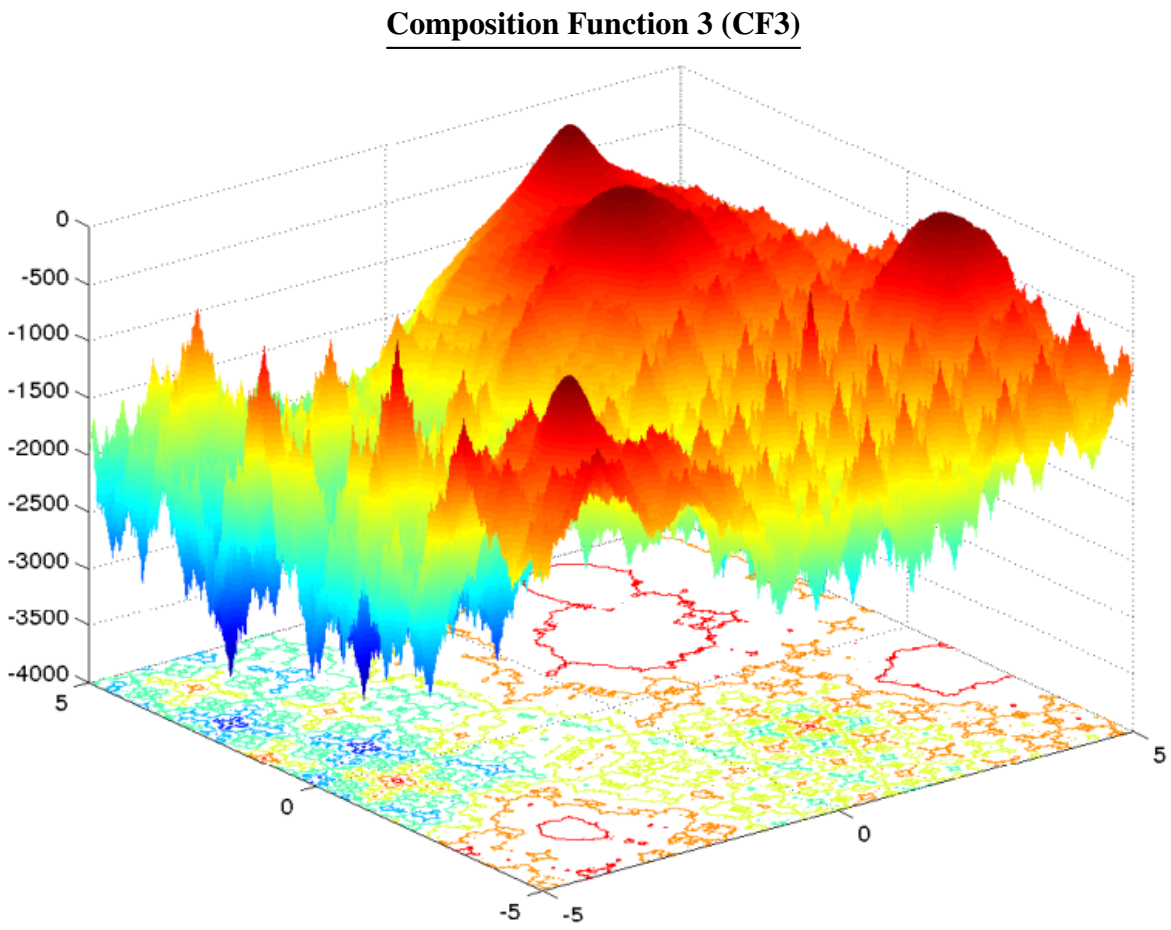


Figure 4.25: Composition Function 3 with fitness on the vertical axis. CF3 is constructed based on six basic functions (2 × EF8F2 function, 2 × Weierstrass function, and 2 × Griewank function), yielding a total of six global optima in the range of $A_D = [5; 5]^D$, here normalised to 0–1. Figure taken from [95].

Chapter 5

Discussion

This thesis focused on the use of genetic algorithms (GAs) to explore complex feature spaces, which are feature spaces for which a systematic exploration would not be practical either due to the size of the space, the complexity required to find solutions, or the need to only find solutions of a certain quality. More specifically, this thesis explored the application of MAP-Elites-style algorithms [118], in which the search of the GA is focused on the illumination of complex feature spaces by encouraging diversity within the population produced (thus, their being termed "illuminative algorithms"). In Chapter 2, we showed the effectiveness of a MAP-Elites-style GA in tackling the exploration of the complex space of possible network structures sharing a specific degree distribution and global clustering coefficient.

There, the search space was defined by encoding network structures in terms of population counts of an arbitrary family of subgraphs (from which networks could be realised via the cardinality matching algorithm (CMA) method; see Sections 1.2 and 2.3.1). This search space was then discretised into cells of size 1, in order to allow the maximum diversity of solutions stored in the population, and optimised towards the desired degree distribution and global clustering coefficient (referred to as a "valid" solution).

The results revealed significant differences in the range of valid solutions compared to state of the art methods (using Dk 2.1 and BigV rewiring in the range of subgraph populations). Further, these differences were shown to have a significant effect on the behaviours of two well-known epidemic simulations, the susceptible-infected-recovered (SIR) and complex contagion simulations (see Section 2.4.3). These results thus demonstrate the presence of a possible bias in current state-of-the-art methods. They further suggest that this kind of GA method of exploration may provide a greater diversity of networks, which can be used to test hypotheses regarding the independence of any studied effect on degree distribution and global clustering.

However, this method has the limitation of being very slow at generating valid solutions, because it optimises solutions through random additions/subtractions to the subgraph encoding (see Section 2.3.1). This does not take into consideration the interconnected relationship between subgraphs, which must be preserved to produce valid, graphical networks. In addition, the extremely low cell size used in this method means that there is no optimisation of solutions within the population, since with such a small cell size all solutions are stored, even if they are not valid.

The size of the cells used could easily be increased to improve the optimisation, and thus the speed at which valid solutions are generated. However, without prior knowledge about the density of valid solutions in the space, this could drastically decrease the diversity of valid solutions found. Specifically, if cells are too large, we risk representing a range of structurally diverse solutions with a single cell.

These limitations were the focus of Chapter 3. There, we developed and demonstrated three extensions to the method intended to address these limitations. First, we focused on improving the efficiency of moving around the space by developing a new method for specifying mutations, which is based on solving a system of Diophantine equations (see Section 3.3.2). This enables movement within the space via valid networks only, which

therefore drastically reduces the computational cost of exploring spaces. This method enables a significantly greater number of valid solutions to be found compared to the random mutations used in Chapter 2. Furthermore, it also increases the level of diversity found in the population. However, due to the methods of encoding and of network realisation used, these results are limited in the range of global clustering values that can be controlled for in the generated networks. That is, this method is only appropriate for clustering levels below the amount that would require the control of clustering-inducing subgraphs that share edges. This limitation is further detailed in Section 3.3.2, but was not the focus of the work presented here. As such, the exploration of alternative methods of network encoding, which still allow control over the higher-order structure of the network, should be the focus of future work to allow a fuller exploration of the space of possible networks. This might even still involve the application of ARC combined with the use of an measure of interestingness (MoI) focusing on those areas of the space that are the hardest to realise for the selected method (i.e., a reverse curiosity measure in which the cells with the least number of revisits are split).

Additionally, the effect of the choice of subgraph families used to encode the networks was not explored in either Chapter 2 or Chapter 3). Whilst these subgraphs were chosen for their relationship to the controlled structural measures (see Section 3.3.1), this would be an interesting parameter to modify to study the importance of particular subgraphs on network structure. This could even be used as an additional controlling factor on the types of network explored, in order to study the dependence of relevant motifs on the dynamics in which they are known to appear. Furthermore, the exploration of this method with alternative target structural measures has not been explored in this thesis. This could require a substantial change to how mutations are formulated, and the choice of subgraphs would have to be re-examined in order to control for the suggested target structural measures.

The second and third additions to the method described in Chapter 2 focused on improving

the exploration of the search space via the development of an adaptive mechanism for setting the resolution of the search, referred to as the adaptive resolution change (ARC) method, and a principle for tuning the size of mutations to the changing resolution of the search. The key idea behind ARC is to resize cells throughout the space to better reflect the level of interestingness (the MoI, here defined as the variation in betweenness centrality, BC) found across the search space. such that the smallest cells reflect the highest levels of MoI found during the search (see Section 3.3.3). These changes in cell size across the space are paired with proportional changes in the size of mutations applied to the individuals within those cells. This means that as the cell size decreases (reflecting the levels of MoI found), the scale of the search in those areas also decreases, becoming more localised around the areas of highest MoI (see Section 3.3.3).

With these extensions, we found that the range of networks structures discovered was significantly increased compared to that generated using the various fixed cell sizes tested. Furthermore, compared to network structures generated by the current state-of-the-art methods (Dk 2.1 and BigV rewiring), we were able to show evidence for sampling bias in these methods by generating valid networks with a completely different range of mean BC. We found that the distribution of solutions provided by these methods is unable to cover the full range of solutions (contradicting claims made in [127]). The additional structural diversity of the discovered networks was then shown to have a significant effect on the behaviour not only of the complex contagion dynamic presented in the previous chapter, but also of the Kuramoto simulation.

Of course, these results are likely to be highly dependent on the type of MoI used. Therefore, the choice of MoI should be a key focus of any further work using this method to explore the effect of other structural variations. This might also involve the exploration of the effectiveness of this method for alternative types of networks (i.e., other degree distributions), which is fully supported by the CMA method of realisation used here.

Furthermore, this method could be expanded to examine directed or even weighted networks, with the weight of edges being encoded and optimised along with the subgraph population. This would require an additional new genome encoding and an associated mutation method (see examples of GAs in the exploration of the weights and topology of networks [135, 83]). With the development of more robust ways of using MAP-Elites like methods within noisy spaces [74], there is no reason that a more direct exploration of the effect of subgraphs on the behavioural dynamics used (i.e., the use of dynamics complex contagion and/or Kuramoto simulations) could not be used as fitness or mapped dimensions in further work. However, this might involve large computational costs per evaluation and thus may require further adaptations to the ARC method (e.g., the use of surrogate fitness like score for the MoI [53, 60, 69]). Both of these methods ([74] and [53, 60, 69]) should be compatible with our proposed ARC method and would be very interesting future topics of research.

In the final chapter (Chapter 4), we focused on the developing the ARC method into a general framework applicable to the exploration of any problem space; more specifically, we investigated the relationship between a given MoI and the mapped space of diversity in a MAP-Elites-like GA. To do so, we expanded and explored the sensitivity of the method to its controlling variables (LL, NN, and Ro), detailing their effect on a new MoI defined in terms of the variance in fitness within a cell.

We then demonstrated the effectiveness of the ARC method for exploring the effect of mapped dimensions on fitness in three distinct cases by comparing it to the standard MAP-Elites method with varying starting cell sizes: (1) three 2D landscapes from the CEC'2013 test suite, designed for comparing niche finding (see Section 4.3.1); (2) exploration of the space of network structures sharing a set degree distribution and global clustering coefficient (as was first discussed in Chapter 3); and (3) a hexapod walking controller, which is commonly used with the MAP-Elites method to create a high-performing and diverse repertoire of behaviours (see Section 4.3.3). In all cases, we were able to show

that the ARC method was able to adjust the resolution of cells across the space to reflect the full range of known niches within the space. In addition, it was able to identify underlying relationships between the mapping dimensions and the MoI, without prior knowledge of the space. Furthermore in all cases explored, the ARC method showed greater coverage than the MAP-Elites method, although with a significantly lower overall optimisation of fitness in the cases in which fitness was used (i.e., the three niche landscapes and the hexapod walking simulations). It should be stressed that the aim of the ARC method presented here was not to optimise fitness, but rather to provide a richer level of detail on the relationship between each of the mapped dimensions and their level of performance and/or the MoI employed. As such, there are many modifications that could be made to the ARC method to improve the level of optimisation within these landscapes, perhaps even without significant decrease to the coverage found here. We lay out examples of the changes that could be made in Section 4.4, including the alteration of the controlling variables of the change events (LL, Ro, and NN) and the selection of a more optimisation-focused MoI, as the MoI used the work done here was highly unsuited to focusing on the areas of highest fitness within the space. Furthermore, we highlight many of the existing extensions to the MAP-Elites method for improving optimisation ability (such as the CVT [174] and CMA-ME [50] methods) and discuss the small alterations that could be made to employ them with the ARC method.
It is worth noting that at the time of running these simulations, we were unaware of attempts to create benchmark landscapes for comparing different MAP-Elites algorithms by [20]. Importantly, the landscapes suggested in [20] were intended to test quality diversity (QD) related performance (i.e., the search for a population of both highly diverse and high-performing solutions), rather than to reflect the underlying relationships in the space, as we focused on here. Nevertheless, a replication of the effectiveness of the ARC method on the Rastrigin function suggested in [20] could serve as basis for comparing the ARC method to a wide range of QD algorithms. Furthermore, the case studies given here, as well as the

examples studied in Chapter 3, are are not meant as an exhaustive survey of the problem space suitable for exploration using the ARC method. Future work should thus start by exploring problems for which domain specific knowledge is lacking, such as antenna design. Furthermore, in the field of QD algorithms, to which the first illumination algorithm (MAP-Elites) now belongs, there is a growing movement away from their illumination roots (i.e., in which informing the space of solutions is the main aim, even at the cost of optimisation). This move makes sense in the kind of problem spaces in which these newer QD algorithms are employed, such as evolutionary robotics [29], neural network design [72, 163], or even game level design [48, 5]. Indeed in these kind of problem spaces, although the choice of measures used for mapping the space (the mapping dimensions), and thus the measures for which diversity is encouraged and or perceived, are carefully considered, this is normally only relates to how they will convey beneficial qualities to the end population [142]. This might be, for example, by providing greater robustness to damage in the case of building a repertoire of behaviours for robot control [41, 31] or by providing benefit as "stepping stones" to regions of higher fitness within the space, which would otherwise be overlooked [148, 23]. However in all of these cases, how the space of solutions is distributed or how each mapped dimension affects the resulting behaviour is overlooked. This is highlighted in [28], in which the mapped dimensions are determined using a neural network method of dimensionality reduction. The search in this work focuses on developing a full range of skills in a robot arm, but at the cost of removing most of the information regarding how the chosen mapped dimension effects relate to the landscapes. However, this is not applicable of all problems and particularly those problems using GAs, in which a lack of detailed knowledge regarding how particular features relate to performance often means that GA-generated solutions vastly outperform current handcrafted solutions, such as in the material sciences [57, 107, 65], in drug discovery [183, 103], and in many other areas [85]. These kinds of "black-box" methods yield

high-fitness solutions, but fail to provide the kind of information crucial to advancing research on these problems. Thus, they cannot provide any kind of transformational creativity [14] to the problem field that might lead to completely different features of interest or even paradigm shifts in the way we look at these problem spaces. The exploration of these kinds of black-box solution spaces by deploying the ARC MAP-Elites method would be a natural next step for this work. This should therefore be the focus of future research, with the aim of adding to domain-specific knowledge of these problem spaces.

Bibliography

[1] Juan A Acebrón, Luis L Bonilla, Conrad J Pérez Vicente, Félix Ritort, and Renato Spigler. The kuramoto model: A simple paradigm for synchronization phenomena. *Reviews of modern physics*, 77(1):137, 2005.

[2] Charu C Aggarwal. An introduction to social network data analytics. In *Social network data analytics*, pages 1–15. Springer, 2011.

[3] Noga Alon, Chen Avin, Michal Kouckỳ, Gady Kozma, Zvi Lotker, and Mark R Tuttle. Many random walks are faster than one. *Combinatorics, Probability and Computing*, 20(4):481–502, 2011.

[4] Uri Alon. Network motifs: theory and experimental approaches. *Nature Reviews Genetics*, 8(6):450, 2007.

[5] Alberto Alvarez, Steve Dahlskog, Jose Font, and Julian Togelius. Empowering quality diversity in dungeon design with interactive constrained map-elites. *arXiv preprint arXiv:1906.05175*, 2019.

[6] Luis A Nunes Amaral and Roger Guimera. Complex networks: Lies, damned lies and statistics. *Nature Physics*, 2(2):75, 2006.

[7] Joshua E Auerbach, Giovanni Iacca, and Dario Floreano. Gaining insight into quality diversity. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*, pages 1061–1064. ACM, 2016.

[8] Nora Ayanian. Dart: Diversity-enhanced autonomy in robot teams. *The International Journal of Robotics Research*, 38(12-13):1329–1337, 2019.

[9] Shweta Bansal, Shashank Khandelwal, and Lauren Ancel Meyers. Exploring biological network structure with clustered random networks. *BMC bioinformatics*, 10(1):405, 2009.

[10] Alain Barrat, Marc Barthelemy, and Alessandro Vespignani. *Dynamical processes on complex networks*. Cambridge university press, 2008.

[11] Kevin E Bassler, Charo I Del Genio, Péter L Erdős, István Miklós, and Zoltán Toroczkai. Exact sampling of graphs with prescribed degree correlations. *New Journal of Physics*, 17(8):083052, 2015.

[12] Richard F Betzel and Danielle S Bassett. Generative models for network neuroscience: prospects and promise. *Journal of The Royal Society Interface*, 14(136):20170623, 2017.

[13] Kurt Binder, Dieter Heermann, Lyle Roelofs, A John Mallinckrodt, and Susan McKay. Monte carlo simulation in statistical physics. *Computers in Physics*, 7(2):156–157, 1993.

[14] Margaret Boden. Chapter thirteen. creativity: How does it work? In *The idea of creativity*, pages 235–250. Brill, 2009.

[15] Stefan Bornholdt and Dirk Graudenz. General asymmetric neural networks and structure design by genetic algorithms. *Neural networks*, 5(2):327–334, 1992.

[16] Michael Breakspear, Stewart Heitmann, and Andreas Daffertshofer. Generative models of cortical oscillations: neurobiological implications of the kuramoto model. *Frontiers in human neuroscience*, 4:190, 2010.

[17] Nathan Brown, Ben McKay, François Gilardoni, and Johann Gasteiger. A graph-based genetic algorithm and its application to the multiobjective evolution of median molecules. *Journal of chemical information and computer sciences*, 44(3):1079–1087, 2004.

[18] J-P Bruneton, L Cazenille, A Douin, and V Reverdy. Exploration and exploitation in symbolic regression using quality-diversity and evolutionary strategies algorithms. *arXiv preprint arXiv:1906.03959*, 2019.

[19] Z Burda, A Krzywicki, OC Martin, and M Zagorski. Motifs emerge from function in model gene regulatory networks. *Proceedings of the National Academy of Sciences*, 108(42):17263–17268, 2011.

[20] Leo Cazenille. Comparing reliability of grid-based quality-diversity algorithms using artificial landscapes. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 249–250, 2019.

[21] Sherry Chalotra, Sumeet Kaur Sehra, and Sukhjit Singh Sehra. An analytical review of nature inspired optimization algorithms. *International Journal of Science Technology & Engineering*, 2(3):123–126, 2015.

[22] Nikhil Chopra, Mark W Spong, and Rogelio Lozano. Synchronization of bilateral teleoperators with time delay. *Automatica*, 44(8):2142–2148, 2008.

[23] Jeff Clune, Jean-Baptiste Mouret, and Hod Lipson. The evolutionary origins of modularity. *Proceedings of the Royal Society b: Biological sciences*, 280(1755):20122863, 2013.

[24] Vittoria Colizza, Alessandro Flammini, M Angeles Serrano, and Alessandro Vespignani. Detecting rich-club ordering in complex networks. *Nature physics*, 2(2):110–115, 2006.

[25] Harry Collins and Luc Berthouze. Optimised Storage for High-Dimensional Multi-Resolution Search Declaration. 2017.

[26] Pol Colomer-de Simón, M Angeles Serrano, Mariano G Beiró, J Ignacio Alvarez-Hamelin, and Marián Boguná. Deciphering the global organization of clustering in real complex networks. *Scientific reports*, 3:2517, 2013.

[27] David Roxbee Cox and David R Cox. *Planning of experiments*, volume 20. Wiley New York, 1958.

[28] Antoine Cully. Autonomous skill discovery with quality-diversity and unsupervised descriptors. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 81–89, 2019.

[29] Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. Robots that can adapt like animals. *Nature*, 521(7553):503–507, 2015.

[30] Antoine Cully and Yiannis Demiris. Quality and diversity optimization: A unifying modular framework. *IEEE Transactions on Evolutionary Computation*, 22(2):245–259, 2017.

[31] Antoine Cully and Yiannis Demiris. Hierarchical behavioral repertoires with unsupervised descriptors. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 69–76, 2018.

[32] Leon Danon, Ashley P Ford, Thomas House, Chris P Jewell, Matt J Keeling, Gareth O Roberts, Joshua V Ross, and Matthew C Vernon. Networks and the epidemiology of infectious disease. *Interdisciplinary perspectives on infectious diseases*, 2011, 2011.

[33] Charles Darwin. *On the Origin of Species by Means of Natural Selection Or the Preservation of Favoured Races in the Struggle for Life.* H. Milford; Oxford University Press, 1869.

[34] Matthias Dehmer. *Structural analysis of complex networks*. Springer Science & Business Media, 2010.

[35] Charo I Del Genio, Hyunju Kim, Zoltán Toroczkai, and Kevin E Bassler. Efficient and exact sampling of simple graphs with given arbitrary degree sequence. *PloS one*, 5(4), 2010.

[36] Emily Dolson, Wolfgang Banzhaf, and Charles Ofria. Applying ecological principles to genetic programming. In *Genetic Programming Theory and Practice XV*, pages 73–88. Springer, 2018.

[37] Emily Dolson, Alexander Lalejini, and Charles Ofria. Exploring genetic programming systems with map-elites. In *Genetic Programming Theory and Practice XVI*, pages 1–16. Springer, 2019.

[38] Stephane Doncieux and Jean-Baptiste Mouret. Behavioral diversity measures for evolutionary robotics. In *IEEE congress on evolutionary computation*, pages 1–8. IEEE, 2010.

[39] Stephane Doncieux and Jean-Baptiste Mouret. Beyond black-box optimization: a review of selective pressures for evolutionary robotics. *Evolutionary Intelligence*, 7(2):71–93, 2014.

[40] Henry Dorrian, Jon Borresen, and Martyn Amos. Community structure and multi-modal oscillations in complex networks. *PloS one*, 8(10), 2013.

[41] Miguel Duarte, Jorge Gomes, Sancho Moura Oliveira, and Anders Lyhne Christensen. Evolution of repertoire-based control for robots with complex locomotor systems. *IEEE Transactions on Evolutionary Computation*, 22(2):314–328, 2017.

[42] Ken TD Eames. Modelling disease spread through random and regular contacts in clustered populations. *Theoretical population biology*, 73(1):104–111, 2008.

[43] Victor M Eguiluz, Dante R Chialvo, Guillermo A Cecchi, Marwan Baliki, and A Vania Apkarian. Scale-free brain functional networks. *Physical review letters*, 94(1):018102, 2005.

[44] Matthias Ehrgott. *Multicriteria optimization*, volume 491. Springer Science & Business Media, 2005.

[45] Agoston E Eiben and Selmar K Smit. Parameter tuning for configuring and analyzing evolutionary algorithms. *Swarm and Evolutionary Computation*, 1(1):19–31, 2011.

[46] Iztok Fister Jr, Xin-She Yang, Iztok Fister, Janez Brest, and Dušan Fister. A brief review of nature-inspired algorithms for optimization. *arXiv preprint arXiv:1307.4186*, 2013.

[47] Dario Floreano and Claudio Mattiussi. *Bio-inspired artificial intelligence: theories, methods, and technologies*. MIT press, 2008.

[48] Matthew C Fontaine, Scott Lee, LB Soros, Fernando De Mesentier Silva, Julian Togelius, and Amy K Hoover. Mapping hearthstone deck spaces through map-elites with sliding boundaries. In *Proceedings of The Genetic and Evolutionary Computation Conference*, pages 161–169, 2019.

[49] Matthew C Fontaine, Scott Lee, Lisa B Soros, Fernando De Mesentier Silva, Julian Togelius, and Amy K Hoover. Mapping hearthstone deck spaces through map-elites with sliding boundaries. In *Proceedings of The Genetic and Evolutionary Computation Conference*, pages 161–169, 2019.

[50] Matthew C Fontaine, Julian Togelius, Stefanos Nikolaidis, and Amy K Hoover. Covariance matrix adaptation for the rapid illumination of behavior space. *arXiv preprint arXiv:1912.02400*, 2019.

[51] David V Foster, Jacob G Foster, Peter Grassberger, and Maya Paczuski. Clustering drives assortativity and community structure in ensembles of networks. *Physical Review E*, 84(6):066117, 2011.

[52] LC Freeman. A set of measures of centrality based on betweenness. sociometry40, 35–41, 1977.

[53] Adam Gaier, Alexander Asteroth, and Jean-Baptiste Mouret. Data-efficient exploration, optimization, and modeling of diverse designs through surrogate-assisted illumination. *Proceedings of the Genetic and Evolutionary Computation Conference on - GECCO17*, 2017.

[54] Ioannis Giagkiozis, Robin C Purshouse, and Peter J Fleming. An overview of population-based algorithms for multi-objective optimisation. *International Journal of Systems Science*, 46(9):1572–1599, 2015.

[55] Daniel Glasscock. a graphon? *Notices of the AMS*, 62(1), 2015.

[56] Daoxiong Gong, Jie Yan, and Guoyu Zuo. A review of gait optimization based on evolutionary computation. *Applied Computational Intelligence and Soft Computing*, 2010, 2010.

[57] Sotirios K Goudos, Christos Kalialakis, and Raj Mittra. Evolutionary algorithms applied to antennas and propagation: A review of state of the art. *International Journal of Antennas and Propagation*, 2016, 2016.

[58] Daniele Gravina, Ahmed Khalifa, Antonios Liapis, Julian Togelius, and Georgios N Yannakakis. Procedural content generation through quality diversity. In *2019 IEEE Conference on Games (CoG)*, pages 1–8. IEEE, 2019.

[59] Darren M Green and Istvan Z Kiss. Large-scale properties of clustered networks: Implications for disease dynamics. *Journal of biological dynamics*, 4(5):431–445, 2010.

[60] Alexander Hagg. Hierarchical surrogate modeling for illumination algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1407–1410, 2017.

[61] George Havas, Bohdan S Majewski, and Keith R Matthews. Extended gcd and hermite normal form algorithms via lattice basis reduction. *Experimental Mathematics*, 7(2):125–136, 1998.

[62] J. H. Holland. An efficient genetic algorithm for the traveling salesman problem. *European Journal of Operational Research*, 606-617., 1975.

[63] JH Holland. An introductory analysis with applications to biology, control, and artificial intelligence. *Adaptation in Natural and Artificial Systems. First Edition, The University of Michigan, USA*, 1975.

[64] Tzung-Pei Hong, Hong-Shung Wang, and Wei-Chou Chen. Simultaneously applying multiple mutation operators in genetic algorithms. *Journal of heuristics*, 6(4):439–455, 2000.

[65] Gregory S Hornby, Jason D Lohn, and Derek S Linden. Computer-automated evolution of an x-band antenna for nasaś space technology 5 mission. *Evolutionary computation*, 19(1):1–23, 2011.

[66] Thomas House, Geoffrey Davies, Leon Danon, and Matt J Keeling. A motif-based approach to network epidemics. *Bulletin of Mathematical Biology*, 71(7):1693–1706, 2009.

[67] Thomas House and Matt J Keeling. The impact of contact tracing in clustered populations. *PLoS computational biology*, 6(3), 2010.

[68] Thomas House and Matt J Keeling. Insights from unifying modern approximations to infections on networks. *Journal of The Royal Society Interface*, 8(54):67–73, 2011.

[69] Md Monjurul Islam, Hemant Kumar Singh, and Tapabrata Ray. A surrogate assisted approach for single-objective bilevel optimization. *IEEE Transactions on Evolutionary Computation*, 21(5):681–696, 2017.

[70] Istvan Z. Kiss, Joel C. Miller, and P ÌĄ eter L. Simon. Mathematics of epidemics on networks: from exact to approximate models-Monograph. *Cham: Springer*, 598, 2016.

[71] Shalev Itzkovitz, Ron Milo, Nadav Kashtan, Guy Ziv, and Uri Alon. Subgraphs in random networks. *Physical review E*, 68(2):026127, 2003.

[72] Ethan C Jackson and Mark Daley. Novelty search for deep reinforcement learning policy network weights by action sequence edit metric distance. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 173–174, 2019.

[73] Niels Justesen, Miguel Gonzalez Duque, Daniel Cabarcas Jaramillo, Jean-Baptiste Mouret, and Sebastian Risi. Learning a behavioral repertoire from demonstrations. *arXiv preprint arXiv:1907.03046*, 2019.

[74] Niels Justesen, Sebastian Risi, and Jean-Baptiste Mouret. Map-elites for noisy domains by adaptive sampling. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 121–122, 2019.

[75] Arpan Kumar Kar. Bio inspired computing–a review of algorithms and scope of applications. *Expert Systems with Applications*, 59:20–32, 2016.

[76] Giorgos Karafotias, Mark Hoogendoorn, and Ágoston E Eiben. Parameter control in evolutionary algorithms: Trends and challenges. *IEEE Transactions on Evolutionary Computation*, 19(2):167–187, 2014.

[77] Brian Karrer and Mark EJ Newman. Random graphs containing arbitrary distributions of subgraphs. *Physical Review E*, 82(6):066118, 2010.

[78] Rituraj Kaushik, Pierre Desreumaux, and Jean-Baptiste Mouret. Adaptive prior selection for repertoire-based online learning in robotics. *arXiv preprint arXiv:1907.07029*, 2019.

[79] Ahmed Khalifa, Scott Lee, Andy Nealen, and Julian Togelius. Talakat: Bullet hell generation through constrained map-elites. In *Proceedings of The Genetic and Evolutionary Computation Conference*, pages 1047–1054, 2018.

[80] Hyunju Kim, Zoltán Toroczkai, Péter L Erdős, István Miklós, and László A Székely. Degree-based graph construction. *Journal of Physics A: Mathematical and Theoretical*, 42(39):392001, 2009.

[81] Manfred G Kitzbichler, Marie L Smith, Søren R Christensen, and Ed Bullmore. Broadband criticality of human brain network synchronization. *PLoS Comput Biol*, 5(3):e1000314, 2009.

[82] Hendrike Klein-Hennig and Alexander K Hartmann. Bias in generation of random graphs. *Physical Review E*, 85(2):026101, 2012.

[83] T Kocmánek. Hyperneat and novelty search for image recognition. *Diss. MasterâĂŹs thesis, Czech Technical University in Prague*, 2015.

[84] Abdullah Konak, David W Coit, and Alice E Smith. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety*, 91(9):992–1007, 2006.

[85] John R Koza, Martin A Keane, Matthew J Streeter, William Mydlowec, Jessen Yu, and Guido Lanza. *Genetic programming IV: Routine human-competitive machine intelligence*, volume 5. Springer Science & Business Media, 2006.

[86] Hugo Kubinyi. Combinatorial and computational approaches in structure-based drug design. *Current Opinion in Drug Discovery and Development*, 1(1):16–27, 1998.

[87] Manoj Kumar, Mohamed Husain, Naveen Upreti, and Deepti Gupta. Genetic algorithm: Review and application. *Available at SSRN 3529843*, 2010.

[88] Yoshiki Kuramoto. Cooperative dynamics of oscillator communitya study based on lattice of rings. *Progress of Theoretical Physics Supplement*, 79:223–240, 1984.

[89] Yoshiki Kuramoto. *Chemical oscillations, waves, and turbulence*. Courier Corporation, 2003.

[90] Ibrahim Kusçu and Chris Thornton. Design of artificial neural networks using genetic algorithms: Review and prospect. 1994.

[91] Pedro Larranaga, Cindy M. H. Kuijpers, Roberto H. Murga, Inaki Inza, and Sejla Dizdarevic. Genetic algorithms for the travelling salesman problem: A review of representations and operators. *Artificial Intelligence Review*, 13(2):129–170, 1999.

[92] Joel Lehman and Kenneth O Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 19(2):189–223, 2011.

[93] Joel Lehman and Kenneth O Stanley. Evolving a diversity of virtual creatures through novelty search and local competition. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 211–218, 2011.

[94] Joel Lehman and Kenneth O Stanley. Novelty search and the problem with objectives. In *Genetic programming theory and practice IX*, pages 37–56. Springer, 2011.

[95] Xiaodong Li, Andries Engelbrecht, and Michael G Epitropakis. Benchmark functions for cecâĂŹ2013 special session and competition on niching methods for multimodal function optimization. *RMIT University, Evolutionary Computation and Machine Learning Group, Australia, Tech. Rep*, 2013.

[96] Xiaodong Li, Michael G Epitropakis, Kalyanmoy Deb, and Andries Engelbrecht. Seeking multiple solutions: an updated survey on niching methods and their applications. *IEEE Transactions on Evolutionary Computation*, 21(4):518–538, 2016.

[97] Gunar E Liepins and Michael D Vose. Deceptiveness and genetic algorithm dynamics. In *Foundations of genetic algorithms*, volume 1, pages 36–50. Elsevier, 1991.

[98] Wen-Yang Lin, Wen-Yung Lee, and Tzung-Pei Hong. Adapting crossover and mutation rates in genetic algorithms. *J. Inf. Sci. Eng.*, 19(5):889–903, 2003.

[99] Hai-Lin Liu, Lei Chen, Kalyanmoy Deb, and Erik D Goodman. Investigating the effect of imbalance between convergence and diversity in evolutionary multiobjective algorithms. *IEEE Transactions on Evolutionary Computation*, 21(3):408–425, 2016.

[100] Fernando G Lobo and Cláudio F Lima. A review of adaptive population sizing schemes in genetic algorithms. In *Proceedings of the 7th annual workshop on Genetic and evolutionary computation*, pages 228–234, 2005.

[101] László Lovász. *Large networks and graph limits*, volume 60. American Mathematical Soc., 2012.

[102] László Lovász, Mihály Bárász, Károly Bezdek, Károly Böröczky, Balázs Csikós, Endre Csóka, György Elekes, Szabolcs Fancsali, András Gács, János Geleji, et al. Diszkrét és folytonos: a gráfelmélet, algebra, analízis és geometria találkozási pontjai= discrete and continuous: interfaces between graph theory, algebra, analysis and geometry. *OTKA Kutatási Jelentések| OTKA Research Reports*, 2012.

[103] Haiping Ma, Minrui Fei, and Zhile Yang. Biogeography-based optimization for identifying promising compounds in chemical process. *Neurocomputing*, 174:494–499, 2016.

[104] Shingo Mabu, Kotaro Hirasawa, and Jinglu Hu. A graph-based evolutionary algorithm: Genetic network programming (gnp) and its extension using reinforcement learning. *Evolutionary computation*, 15(3):369–398, 2007.

[105] DJ Maddalena and GM Snowdon. Applications of genetic algorithms to drug design. *Expert Opinion on Therapeutic Patents*, 7(3):247–254, 1997.

[106] Katherine M Malan and Andries P Engelbrecht. A survey of techniques for characterising fitness landscapes and some possible ways forward. *Information Sciences*, 241:148–163, 2013.

[107] Diogenes Marcano and Filinto Durán. Synthesis of antenna arrays using genetic algorithms. *IEEE Antennas and Propagation Magazine*, 42(3):12–20, 2000.

[108] Patrick N McGraw and Michael Menzinger. Clustering and the synchronization of oscillator networks. *Physical Review E*, 72(1):015101, 2005.

[109] David Mears and Harvey B Pollard. Network science and the human brain: using graph theory to understand the brain and one of its hubs, the amygdala, in health and disease. *Journal of neuroscience research*, 94(6):590–605, 2016.

[110] Christos Gkantsidist Milena Mihail and Ellen Zegura. The markov chain simulation method for generating connected power law random graphs. In *Proceedings of the Fifth Workshop on Algorithm Engineering and Experiments*, volume 111, page 16. SIAM, Philadelphia, 2003.

[111] Joel C Miller. Complex contagions and hybrid phase transitions in unclustered and clustered random networks. *arXiv preprint arXiv:1501.01585*, 2015.

[112] Ron Milo, Nadav Kashtan, Shalev Itzkovitz, Mark EJ Newman, and Uri Alon. On the uniform generation of random graphs with prescribed degree sequences. *arXiv preprint cond-mat/0312028*, 2003.

[113] Ron Milo, Shai Shen-Orr, Shalev Itzkovitz, Nadav Kashtan, Dmitri Chklovskii, and Uri Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.

[114] Melanie Mitchell. *An introduction to genetic algorithms*. MIT press, 1998.

[115] Adilson E Motter and Zoltan Toroczkai. Introduction: Optimization in networks, 2007.

[116] J-B Mouret and Stéphane Doncieux. Encouraging behavioral diversity in evolutionary robotics: An empirical study. *Evolutionary computation*, 20(1):91–133, 2012.

[117] Jean-Baptiste Mouret. Novelty-based multiobjectivization. In *New horizons in evolutionary robotics*, pages 139–154. Springer, 2011.

[118] Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*, 2015.

[119] M. E J Newman. *Networks: an introduction. Oxford University*. Oxford University Press Inc., New York, 2010.

[120] Mark EJ Newman. Spread of epidemic disease on networks. *Physical review E*, 66(1):016128, 2002.

[121] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[122] Mark EJ Newman. Random graphs with clustering. *Physical review letters*, 103(5):058701, 2009.

[123] Mark EJ Newman, Steven H Strogatz, and Duncan J Watts. Random graphs with arbitrary degree distributions and their applications. *Physical review E*, 64(2):026118, 2001.

[124] Trung Thanh Nguyen, Shengxiang Yang, and Juergen Branke. Evolutionary dynamic optimization: A survey of the state of the art. *Swarm and Evolutionary Computation*, 6:1–24, 2012.

[125] Jeffrey V Nickerson, Yasuaki Sakamoto, and Lixiu Yu. Structures for creativity: The crowdsourcing of design. In *CHI workshop on crowdsourcing and human computation*, pages 1–4. Citeseer, 2011.

[126] Jørgen Nordmoen, Eivind Samuelsen, Kai Olav Ellefsen, and Kyrre Glette. Dynamic mutation in map-elites for robotic repertoire generation. In *Artificial Life Conference Proceedings*, pages 598–605. MIT Press, 2018.

[127] Chiara Orsini, Marija M Dankulov, Pol Colomer-de Simón, Almerima Jamakovic, Priya Mahadevan, Amin Vahdat, Kevin E Bassler, Zoltán Toroczkai, Marián Boguñá, Guido Caldarelli, et al. Quantifying randomness in real networks. *Nature communications*, 6(1):1–10, 2015.

[128] David JP OŚullivan, Gary James OḰeeffe, Peter G Fennell, and James P Gleeson. Mathematical modeling of complex contagion on clustered networks. *Frontiers in Physics*, 3:71, 2015.

[129] Peter Overbury and Luc Berthouze. Using novelty-biased ga to sample diversity in graphs satisfying constraints. In *Proceedings of the Companion Publication of the*

2015 Annual Conference on Genetic and Evolutionary Computation, pages 1445–1446, 2015.

[130] Peter Overbury, István Z Kiss, and Luc Berthouze. A genetic algorithm-based approach to mapping the diversity of networks sharing a given degree distribution and global clustering. In *International Workshop on Complex Networks and their Applications*, pages 223–233. Springer, 2016.

[131] Peter Overbury, István Z Kiss, and Luc Berthouze. Mapping structural diversity in networks sharing a given degree distribution and global clustering: Adaptive resolution grid search evolution with diophantine equation-based mutations. In *International Conference on Complex Networks and their Applications*, pages 718–730. Springer, 2018.

[132] Gary B Parker. Co-evolving model parameters for anytime learning in evolutionary robotics. *Robotics and Autonomous Systems*, 33(1):13–30, 2000.

[133] Romualdo Pastor-Satorras, Claudio Castellano, Piet Van Mieghem, and Alessandro Vespignani. Epidemic processes in complex networks. *Reviews of modern physics*, 87(3):925, 2015.

[134] VP Patil and DD Pawar. The optimal crossover or mutation rates in genetic algorithm: a review. *International Journal of Applied Engineering and Technology*, 5(3):38–41, 2015.

[135] Mantas Paulinas and Andrius Ušinskas. A survey of genetic algorithms applications for image enhancement and segmentation. *Information Technology and control*, 36(3), 2007.

[136] Martin Pelikan and David E Goldberg. Escaping hierarchical traps with competent genetic algorithms. In *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, pages 511–518. Morgan Kaufmann Publishers Inc., 2001.

[137] Ioannis Pitas. *Graph-based social media analysis*, volume 39. CRC Press, 2016.

[138] Mike Preuss. Improved topological niching for real-valued global optimization. In *European Conference on the Applications of Evolutionary Computation*, pages 386–395. Springer, 2012.

[139] Mike Preuss. *Multimodal optimization by means of evolutionary algorithms*. Springer, 2015.

[140] Justin K Pugh, Lisa B Soros, and Kenneth O Stanley. An extended study of quality diversity algorithms. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*, pages 19–20, 2016.

[141] Justin K Pugh, Lisa B Soros, and Kenneth O Stanley. Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI*, 3:40, 2016.

[142] Justin K Pugh, Lisa B Soros, Paul A Szerlip, and Kenneth O Stanley. Confronting the challenge of quality diversity. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 967–974, 2015.

[143] A Ramachandra Rao, Rabindranath Jana, and Suraj Bandyopadhyay. A markov chain monte carlo method for generating random (0, 1)-matrices with given marginals. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 225–242, 1996.

[144] Anatol Rapoport. Cycle distributions in random nets. *The bulletin of mathematical biophysics*, 10(3):145–157, 1948.

[145] Noraini Mohd Razali, John Geraghty, et al. Genetic algorithm performance with different selection strategies in solving tsp. In *Proceedings of the world congress on engineering*, volume 2, pages 1–6. International Association of Engineers Hong Kong, 2011.

[146] Jonathan M Read and Matt J Keeling. Disease evolution on networks: the role of contact structure. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 270(1516):699–708, 2003.

[147] Colin R. Reeves. Evolutionary computation: a unified approach. *Genetic Programming and Evolvable Machines*, 2007.

[148] Sebastian Risi, Charles E Hughes, and Kenneth O Stanley. Adaptive Behavior with novelty search. *Adaptive Behavior*, 2010.

[149] Martin Ritchie, Luc Berthouze, Thomas House, and Istvan Z Kiss. Higher-order structure and epidemic dynamics in clustered networks. *Journal of Theoretical Biology*, 348:21–32, 2014.

[150] Martin Ritchie, Luc Berthouze, and Istvan Z Kiss. Beyond clustering: mean-field dynamics on networks with arbitrary subgraph composition. *Journal of mathematical biology*, 72(1-2):255–281, 2016.

[151] Martin Ritchie, Luc Berthouze, and Istvan Z Kiss. Generation and analysis of networks with a prescribed degree sequence and subgraph family: higher-order structure matters. *Journal of complex networks*, 5(1):1–31, 2017.

[152] Gareth O Roberts. Markov chain concepts related to sampling algorithms. *Markov chain Monte Carlo in practice*, 57:45–58, 1996.

[153] Garry Robins, Pip Pattison, Yuval Kalish, and Dean Lusher. An introduction to exponential random graph (p*) models for social networks. *Social networks*, 29(2):173–191, 2007.

[154] Nitzan Rosenfeld, Michael B Elowitz, and Uri Alon. Negative autoregulation speeds the response times of transcription networks. *Journal of molecular biology*, 323(5):785–793, 2002.

[155] Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *science*, 324(5923):81–85, 2009.

[156] R Sivaraj and T Ravichandran. A review of selection methods in genetic algorithm. *International journal of engineering science and technology*, 3(5):3792–3797, 2011.

[157] Davy Smith, Laurissa Tokarchuk, and Geraint Wiggins. Rapid phenotypic landscape exploration through hierarchical spatial partitioning. In *International conference on parallel problem solving from nature*, pages 911–920. Springer, 2016.

[158] Tom Smith, Phil Husbands, and Michael OŚhea. Fitness landscapes and evolvability. *Evolutionary computation*, 10(1):1–34, 2002.

[159] Sara Nadiv Soffer and Alexei Vazquez. Network clustering coefficient without degree-correlation biases. *Physical Review E*, 71(5):057101, 2005.

[160] Kenneth O Stanley, David D Ambrosio, and Jason Gauci. A Hypercube-Based Indirect Encoding for Evolving Large-Scale Neural Networks. Technical Report 2, 2009.

[161] Kenneth O Stanley and Risto Miikkulainen. A taxonomy for artificial embryogeny. *Artificial Life*, 9(2):93–130, 2003.

[162] Clara Stegehuis, Remco van der Hofstad, and Johan SH van Leeuwaarden. Variational principle for scale-free network motifs. *Scientific reports*, 9(1):1–10, 2019.

[163] Felipe Petroski Such, Vashisht Madhavan, Edoardo Conti, Joel Lehman, Kenneth O Stanley, and Jeff Clune. Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. *arXiv preprint arXiv:1712.06567*, 2017.

[164] Kazuhiro Takemoto, Chikoo Oosawa, and Tatsuya Akutsu. Structure of n-clique networks embedded in a complex network. *Physica A: Statistical Mechanics and its Applications*, 380:665–672, 2007.

[165] Hisashi Tamaki, Hajime Kita, and Shigenobu Kobayashi. Multi-objective optimization by genetic algorithms: A review. In *Proceedings of IEEE international conference on evolutionary computation*, pages 517–522. IEEE, 1996.

[166] Kay Chen Tan, Swee Chiang Chiam, AA Mamun, and Chi Keong Goh. Balancing exploration and exploitation with adaptive variation for evolutionary multi-objective optimization. *European Journal of Operational Research*, 197(2):701–713, 2009.

[167] Danesh Tarapore, Jeff Clune, Antoine Cully, and Jean-Baptiste Mouret. How do different encodings influence the performance of the map-elites algorithm? In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, pages 173–180, 2016.

[168] David M Tate and Alice E Smith. Expected allele coverage and the role of mutation in genetic algorithms. In *ICGA*, volume 31, page 37, 1993.

[169] AJ Umbarkar and PD Sheth. Crossover operators in genetic algorithms: a review. *ICTACT journal on soft computing*, 6(1), 2015.

[170] Neil Urquhart and Emma Hart. Optimisation and illumination of a real-world workforce scheduling and routing application (wsrp) via map-elites. In *International Conference on Parallel Problem Solving from Nature*, pages 488–499. Springer, 2018.

[171] David A Van Veldhuizen and Gary B Lamont. Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. *Evolutionary computation*, 8(2):125–147, 2000.

[172] Vesselin K Vassilev, Terence C Fogarty, and Julian F Miller. Smoothness, ruggedness and neutrality of fitness landscapes: from theory to application. In *Advances in evolutionary computing*, pages 3–44. Springer, 2003.

[173] Vassiiis Vassiliades and Jean-Baptiste Mouret. Discovering the elite hypervolume by leveraging interspecies correlation. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 149–156, 2018.

[174] Vassilis Vassiliades, Konstantinos Chatzilygeroudis, and Jean-Baptiste Mouret. Scaling up map-elites using centroidal voronoi tessellations. *arXiv preprint arXiv:1610.05729*, 2016.

[175] Vassilis Vassiliades, Konstantinos Chatzilygeroudis, and Jean-Baptiste Mouret. Comparing multimodal optimization and illumination. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 97–98, 2017.

[176] Vassilis Vassiliades, Konstantinos Chatzilygeroudis, and Jean-Baptiste Mouret. A comparison of illumination algorithms in unbounded spaces. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1578–1581, 2017.

[177] Vassilis Vassiliades, Konstantinos Chatzilygeroudis, and Jean-Baptiste Mouret. Using centroidal voronoi tessellations to scale up the multidimensional archive of phenotypic elites algorithm. *IEEE Transactions on Evolutionary Computation*, 22(4):623–630, 2017.

[178] Roby Velez and Jeff Clune. Novelty search creates robots with general skills for exploration. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, pages 737–744, 2014.

[179] Erik Volz. Random networks with tunable degree distribution and clustering. *Physical Review E*, 70(5):056115, 2004.

[180] Erik M Volz, Joel C Miller, Alison Galvani, and Lauren Ancel Meyers. Effects of heterogeneous and clustered contact patterns on infectious disease dynamics. *PLoS computational biology*, 7(6), 2011.

[181] Michael D Vose. A closer look at mutation in genetic algorithms. *Annals of Mathematics and Artificial Intelligence*, 10(4):423–434, 1994.

[182] Duncan J Watts and Steven H Strogatz. Collective dynamics of small-world networks. *nature*, 393(6684):440, 1998.

[183] Lutz Weber. Evolutionary combinatorial chemistry: application of genetic algorithms. *Drug Discovery Today*, 3(8):379–385, 1998.

[184] Sewall Wright. Evolution in mendelian populations. *Genetics*, 16(2):97, 1931.

[185] Xin-She Yang. Review of metaheuristics and generalized evolutionary walk algorithm. *arXiv preprint arXiv:1105.3668*, 2011.

[186] Zeping Zhan, Batu Aytemiz, and Adam M Smith. Taking the scenic route: Automatic exploration for videogames. *arXiv preprint arXiv:1812.03125*, 2018.

[187] Aimin Zhou, Bo-Yang Qu, Hui Li, Shi-Zheng Zhao, Ponnuthurai Nagaratnam Suganthan, and Qingfu Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49, 2011.