

PHD

Predicting optimal self-assembly of patchy colloids via simulation **An exploration of local search metaheuristics on a noisy yield landscape**

Wan, Tsimhei

Award date:
2019

Awarding institution:
University of Bath

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Citation for published version:

Wan, T 2018, 'Predicting optimal self-assembly of patchy colloids via simulation: An exploration of local search metaheuristics on a noisy yield landscape', Ph.D., University of Bath.

Publication date:

2018

Document Version

Early version, also known as pre-print

[Link to publication](#)

University of Bath

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Predicting optimal self-assembly of patchy colloids via simulation

An exploration of local search metaheuristics
on a noisy yield landscape

by

Tsimhei Wan

Thesis

Submitted for the degree of
Doctor of Philosophy

University of Bath

Department of Physics

December 2018

Copyright Notice

Attention is drawn to the fact that copyright of this thesis rests with the author. A copy of this thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with the author and that they must not copy it or use material from it except as permitted by law or with the consent of the author.

This thesis may be made available for consultation within the University Library and may be photocopied or lent to other libraries for the purposes of consultation with effect from (date).....

Acknowledgements

I would like to thank my supervisor Prof. Nigel Wilding for his patience and support over the past four years, and for the opportunity to take on this project.

Thank you to all members of 3W 4.5 for the tea and coffee breaks, served with a side of puns; all the best. Thank you to all organisers of events and workshops for the learning opportunities.

To my family, especially Mum and Dad and my beloved grandma I appreciate the unconditional support over the years. To all friends, a good catch up is always welcome. A special mention to Vince for sharing a passion for food and drink, and Harry, my partner in crime best of luck with completing your PhD.

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 1.1 | Thesis Outline | 4 |
| 2 | Colloidal self-assembly and particle design | 5 |
| 2.1 | Colloidal systems | 6 |
| 2.2 | Patchy colloidal particles | 7 |
| 2.3 | Core concepts of colloidal self-assembly | 9 |
| 2.4 | Designing self-assembling systems | 11 |
| 2.4.1 | Targeted design | 12 |
| | Forward design | 12 |
| | Inverse design | 12 |
| | Choosing between the two approaches | 14 |
| 2.4.2 | Parameter tuning | 14 |
| 2.4.3 | Speeding up the exploration and optimisation of parameter values . | 15 |
| 2.4.4 | Speeding up particle simulations | 16 |
| 2.5 | Conclusion | 18 |
| 3 | Monte Carlo simulations | 19 |
| 3.1 | Modelling particle systems | 19 |
| 3.2 | Monte Carlo method | 20 |
| 3.2.1 | Statistical Ensemble | 21 |
| 3.2.2 | Ergodicity and detailed balance | 22 |
| 3.2.3 | Metropolis Monte Carlo | 24 |
| 3.2.4 | Simulating dynamical systems | 25 |
| 3.3 | Virtual-Move Monte Carlo (VMMC) | 25 |
| 3.3.1 | Move proposal | 26 |
| 3.3.2 | Move acceptance | 27 |
| 3.4 | Conclusion | 28 |
| 4 | Self-assembly of a patchy disk system | 31 |
| 4.1 | Patchy disk model | 32 |
| 4.1.1 | Review on this particular model | 32 |
| 4.2 | Simulation details | 35 |
| 4.3 | Measures | 36 |

| | | |
|----------|--|-----------|
| 4.3.1 | Local bond-orientational order and bond-orientational order susceptibility | 38 |
| 4.3.2 | Shape factor (compactness-circularity measure) | 39 |
| | Target cluster—constructing our ideal honeycomb cluster | 40 |
| | Radius of gyration | 42 |
| | COM-based overlap measure | 43 |
| | Coordination number | 44 |
| | Performance | 44 |
| 4.3.3 | Compound yield measure | 45 |
| 4.3.4 | Summary | 47 |
| 4.4 | Self-assembly observations | 47 |
| 4.4.1 | Assembly yield at final time | 47 |
| 4.4.2 | Effects of tuning α | 51 |
| 4.4.3 | Differences between the two dynamical regimes of the VMMC algorithm | 52 |
| 4.4.4 | Closed loop formation | 54 |
| 4.4.5 | Cluster growth dynamics—effects on the cluster morphology | 54 |
| 4.4.6 | Summary | 57 |
| 4.5 | Yield landscape properties | 59 |
| 4.5.1 | Time dependency | 59 |
| 4.5.2 | Noise in the landscape: fluctuations in the measurements | 59 |
| 4.5.3 | Computational demand of VMMC simulations | 61 |
| 4.6 | Conclusion | 61 |
| 5 | Review of local search heuristics and hill-climbing techniques | 63 |
| 5.1 | Optimisation problems | 63 |
| 5.1.1 | Heuristics and metaheuristics | 65 |
| 5.2 | Redefining the parameter tuning problem | 67 |
| 5.2.1 | Constraints in simulation-based parameter tuning | 67 |
| 5.2.2 | Measuring search performance | 68 |
| | Computational demand of VMMC simulations and search | 69 |
| | Approaches to improving performance | 70 |
| 5.3 | Noisy fitness landscape | 71 |
| 5.4 | Hill-climbing on noisy landscapes | 71 |
| 5.4.1 | Neighbourhood | 73 |
| 5.4.2 | Effects of intrinsic noise | 74 |
| 5.4.3 | Details of intrinsic noise effects | 76 |
| 5.4.4 | Reducing noise | 77 |
| 5.4.5 | Alternative approach—reusing past sampled values | 79 |
| 5.5 | Conclusion | 80 |

| | | |
|----------|---|------------|
| 6 | Hill-climbing on self-assembly landscapes | 81 |
| 6.1 | Pre-model studies: direct application of hill-climbing with VMMC simulations | 83 |
| 6.1.1 | Direct application of greedy hill-climbing algorithm with VMMC simulations to tune parameters | 83 |
| 6.1.2 | Observations | 84 |
| 6.2 | Yield landscape model \hat{Q} | 85 |
| 6.2.1 | Yield landscape model construction | 85 |
| 6.2.2 | Comparison of the yield landscape model \hat{Q} with simulation yield data Q | 88 |
| 6.3 | Hill-climbing algorithms | 91 |
| 6.3.1 | Trial period limit m | 91 |
| 6.3.2 | Neighbour selection scheme | 91 |
| 6.3.3 | Cumulative averaging of past sampled values | 92 |
| 6.4 | Hill-climbing observations | 93 |
| 6.4.1 | Fitness changes in hill-climbing | 95 |
| 6.4.2 | Averaged fitness performance for different protocols | 95 |
| 6.4.3 | Influence of m on terminal fitness distribution of climbers | 100 |
| 6.4.4 | Discussion | 100 |
| 6.5 | Move probabilities | 104 |
| 6.5.1 | Discussion | 107 |
| 6.6 | Climbing to threshold fitness | 108 |
| 6.6.1 | Experimental details and results | 109 |
| 6.6.2 | Discussion | 112 |
| 6.7 | One-dimensional noisy hill climbs | 112 |
| 6.7.1 | 1D fitness landscapes model and experimental setup | 113 |
| 6.7.2 | Results | 115 |
| 6.7.3 | Discussion | 115 |
| 6.8 | Revisiting the climbing schedule strategy: searching a changing landscape | 117 |
| 6.8.1 | Setup | 117 |
| 6.8.2 | Results | 118 |
| 6.8.3 | Discussion | 118 |
| 6.9 | Conclusion | 123 |
| 7 | Outlook | 125 |
| 7.1 | Future Work | 127 |

List of Figures

| | | |
|------|--|----|
| 2.1 | Examples of patchy colloidal particles. | 8 |
| 4.1 | Schematic of patchy disk model particle. | 33 |
| 4.2 | Schematic phase diagram highlighting how ε and w influences the equilibrium and non-equilibrium behaviour of three-patched hard disk model system to realise many real physical systems. | 33 |
| 4.3 | Key legend for polygon loops of different sizes, and diagram showing restricted movement disks in non-hexagonal loops. | 34 |
| 4.4 | Snapshots of VMMC simulations and possible outcomes. | 37 |
| 4.5 | Examples of honeycomb cluster networks. | 40 |
| 4.6 | Construction of the ideal honeycomb lattices H^* | 41 |
| 4.7 | Example cluster networks with different COM resulting in different values in COM-based shape factors. | 46 |
| 4.8 | Yield heatmap from VMMC simulations annotated with VMMC simulation snapshots. | 48 |
| 4.9 | Bond-orientational order susceptibility $\chi_{6,C}$ and shape factor N_I/N_C heatmaps at final VMMC time $t_f = 2 \times 10^7$ VMMC cycles. | 50 |
| 4.10 | Cluster size N_C heatmap at final VMMC time $t_f = 2 \times 10^7$ VMMC cycles. | 51 |
| 4.11 | Expected yield Q for various α values. | 53 |
| 4.12 | Total energy magnitude $ E $ heatmap at final VMMC time $t_f = 2 \times 10^7$ VMMC cycles. | 53 |
| 4.13 | Stacked time trajectories of system-wide polygon populations for $w = 8^\circ$ patchy disks at various strengths ε | 55 |
| 4.14 | Stacked time trajectories of system-wide polygon populations dependent on VMMC time t_V | 56 |
| 4.15 | Effect of merging of clusters. | 58 |
| 4.16 | An assortment of yield landscape property plots. | 60 |
| 5.1 | Neighbourhood definition for the yield landscape Q on parameter search space \mathcal{P} | 74 |
| 6.1 | Q trajectory of a hill climb search coupled with VMMC simulation on the final time yield landscape Q_{t_f} , initialised at $\mathbf{x}_0 = (12, 12)$ | 86 |

| | | |
|------|--|-----|
| 6.2 | Footprint of the trajectories of five hill-climbing runs coupled with VMMC simulations on the final time yield landscape Q_{t_f} | 87 |
| 6.3 | Comparison between the yield landscape model \hat{Q} and the simulated yield landscape Q | 89 |
| 6.4 | True fitness $\langle \hat{Q} \rangle$ and measured fitness \hat{Q} trajectories for typical hill-climbing runs for various protocols. | 94 |
| 6.5 | Averaged true fitness \bar{Q} trajectories for different procotols and initial points, with and without cumulative averaging routine switched on. | 98 |
| 6.6 | Plots of terminal climber distribution on \mathcal{P} | 101 |
| 6.7 | Plots of terminal climber distribution on \mathcal{P} | 102 |
| 6.8 | Histogram of the late time true fitness distribution of the climbers initiated at $\mathbf{x} = (8, 6), (6, 19)$ and $(14, 19)$ for $m = 1, 8$ and 1000 | 103 |
| 6.9 | Schematic of the local move probabilities. | 106 |
| 6.10 | Plots of $P(\Delta g > 0 Y^*(\mathbf{x}))$ and $P(\Delta f > 0 \Delta g > 0, Y(\mathbf{x}))$ over a typical hill climb. | 107 |
| 6.11 | Plots of frequency of Y^* and $P(\Delta g > 0, Y(\mathbf{x}))$ over a typical hill climb. . . | 108 |
| 6.12 | Plots of the average number of iterations (NSTEPS); evaluations (NEVALS); trials (NTRIALS); and successful moves (NSUCCS) of a threshold hill climb. . | 109 |
| 6.13 | Plots of relative NEVALS taken to reach threshold fitness \bar{Q}^* for a a range of m from initial point $\mathbf{x}_0 = (10, 19)$ with different evaluation sample sizes r . . | 111 |
| 6.14 | Plots of the six 1D fitness functions F1 to F6 used. | 114 |
| 6.15 | Plots of the relative NEVALS and NSTEPS for the six 1D fitness landscapes. . | 115 |
| 6.16 | Plots of the fitness trajectories of hill climbs on multi-duration yield landscapes \hat{Q}_{mult} as a function of k for different m and κ | 119 |
| 6.17 | Plots of the fitness trajectories of hill climbs on multi-duration yield landscapes as a function of t_{TOTCPU} for different m and κ | 120 |
| 6.18 | Bar charts comparing the underlying average fitness \bar{Q} and the total CPU cost $\langle t_{TOTCPU} \rangle$ for hill climbs performed on multi-duration landscapes. . . | 121 |

List of Tables

| | | |
|-----|--|-----|
| 4.1 | Comparison of measures describing morphology | 45 |
| 4.2 | Best Q state points in the parameter search space for $\alpha = 1.4$ | 52 |
| 6.1 | Average \bar{Q} outcomes for hill climbs on model yield landscape \hat{Q} for different initial points. | 96 |
| 6.2 | Summary of the six 1D noisy fitness landscapes G1 to G6 tested. | 114 |

Abstract

Self-assembly systems, based on polymers, nanoparticles, colloidal sized particles or other derivatives, are an appealing way to create materials with high complexity and detail. These systems are highly sensitive to changes in the interactions between the system's components, leading to different emergent structures. In material science, targeted design of new colloidal systems requires the tuning of interactions such that the desired target structure or macrobehaviour is emergent, and to obtain a high-yielding self-assembly path. Changes to the interactions can be represented by a parameter space. Time-dependent simulations (or experiments) are necessary to find emergent systems in parameter space, and the complexity of them demands resources and time. The conventional systematic (brute-force) scan of parameter space is highly inefficient, with most resources spent evaluating low-yielding regions. Typically, to counter noisy measurements to accurately identify favourable parameter values, an average is taken across multiple simulations. We examine hill-climbing as a potential alternative for tuning parameters values to obtain high-yielding systems. As an example tuning problem, a two-dimensional shorted-ranged attractive patchy hard disk model, important for coarse-grain modelling of polymers and biological systems, and a yield measure for quantifying our target structure (large round compact honeycomb clusters) are introduced. Varying the interaction strength and patch width, a noisy landscape is constructed from Monte Carlo simulation yield data. We show that a hill-climbing search on this landscape can locate the localised region of high-yielding assembly, and suggest situations where this is advantageous over brute-force scan. Additionally, we introduce novel meta-heuristics and strategies to augment hill-climbing (including reusing and cumulative averaging of measurements, and scheduled increase of simulation lengths) to handle noise. Reduction in the demand for computational resources is possible at the cost of increased execution times. This extends the algorithmic protocols available, different from averaging over multiple samples, for controlling the computational cost of search.

List of Symbols

| | |
|-----------------------------|--|
| $\langle \cdot \rangle_t$ | Average of observable at time t |
| α | Shape factor weight parameter |
| Δ_t | Maximum translation step size |
| Δ_r | Maximum rotation step size |
| ε | Interaction strength |
| η | Area fraction |
| κ | Schedule update rate |
| σ | Standard deviation |
| χ_6 | Bond orientational order susceptibility |
| C | Cluster or principal cluster |
| $H_{sym}(q)$ | Symmetric honeycomb clusters with q layers |
| H^* | Ideal honeycomb cluster |
| N | Number of particles |
| N_C | Cluster size |
| N_I | Number of disk centres lying within ideal cluster circle |
| N_I/N_C | Shape factor |
| $\mathcal{N}(x)$ | Neighbourhood of x |
| $\mathcal{N}_{\text{norm}}$ | Normal distribution |
| \mathcal{P} | Parameter space |
| Q | Yield |
| \hat{Q} | Noisy yield landscape model |
| \bar{Q} | True yield fitness landscape of \hat{Q} |
| $S(N_C)$ | Ideal cluster circle for N_C disks |
| \mathcal{S} | Search space |
| T | Temperature |
| Y | Fitness measurement fluctuation |
| a | Particle diameter |
| f | Objective function, true/underlying fitness function/landscape |
| g | Noisy fitness function/landscape |
| k | Iteration |
| k_B | Boltzmann constant |
| k_{max} | Maximum number of iterations—hill-climb length |
| m | Trial period limit |
| r | Distance(particle simulations); evaluation sample size (hill-climbing) |

| | |
|---------------------|---|
| r_{hist} | Maximum sample size of history |
| t_{CPU} | CPU time |
| t_{TOTCPU} | Total CPU time |
| t_{EXEC} | Execution time |
| t_f | Final assembly time/simulation duration |
| t_V | Time in number of VMMC cycles |
| x | Solution/state points |
| \mathbf{x}^* | Yield global maximum solution/state point, best yielding parameters |
| w | Patch flexibility |
| COM | Centre of mass |
| CPU | Central processing unit |
| MC | Monte Carlo |
| SPMC | Single Particle Monte Carlo |
| VMMC | Virtual-move Monte Carlo |
| NEVALS | Number of evaluations |
| NSTEPS | Number of iterations/steps |
| NTRIALS | Number of trials |
| NSUCCS | Number of successful moves |

Chapter 1

Introduction

Material scientists are becoming increasingly interested in nanoscale and mesoscale synthesis of particles for the potential they provide in creating new novel materials. Fully-controlled construction of micro- and nano- scopic structures using techniques such as lithography and ion beams require high precision and expensive specialised equipment [1; 2]. Self-assembly of small colloidal particles have many desirable features that can overcome these limitations. Once the right conditions are set up, a collection of particles, usually in solution, will spontaneously organise themselves into a desired structure. These conditions include temperature, concentration and other environmental parameters that are easy to control. Scaling up these processes thus becomes relatively easier, and specialist apparatus may not be required.

Being able to understand and direct the self-assembly of components into some intended structure would enable greater control over the fabrication of materials. By building ‘pre-programmed’ constituent particles that assemble by themselves into a desired form under given conditions, the monitoring of fabrication processes can be reduced. Clearly this would be advantageous for increasing yield during large industrial scale production of materials such as catalysts [3], as well as offering sensitive functional properties with potential applications in optoelectronics [4; 5] and drug delivery [6–8]. At the same time, this ‘bottom-up’ approach to fabrication enables us to further our current understanding of particle interactions and phase behaviour [9–15], and have potential impact in chemical and biological engineering [16; 17].

We study these phenomena at the molecular level using a combination of statistical mechanics and computer simulations. With many design parameters and parameter values to tune, it is hard to only rely on experimental work to test out and develop colloidal design principles. Modelling particle systems enable us to test colloidal designs and predict their assembly outcomes, screening out poor yielding parameters and parameter values before designs are advanced to the experimental stages.

As progress is being made in understanding self-assembly principles, and synthesis of colloids are becoming more intricate and complex [18–20], our attention is turning towards a targeted design of colloidal particles. The goal is to start with the target structure and/or desired aggregate properties in mind, then tailor the design of the colloids to influence the resulting system such that it exhibits the matching emergent self-assembly.

One major aspect is the tuning of design/interaction parameter values to bring out and enhance the assembly yield of the target structure. Two design approaches have been recognised: *forward* and *inverse* design. As the names suggest, they are viewed as opposite approaches to one another. The forward approach represents the conventional method of selecting a base design informed from theory and past examples before tuning parameters via systematic particle simulations. The inverse approach starts with the target in mind to control decision making, iteratively modifying a functional design to match selected characteristic qualities of the design criteria, such as the radial distribution function or the desired crystalline configuration, to optimise complex multi-parametric models to give better performing solutions.

Although inverse design approaches are hailed as the modern approach to functional design, many of these techniques are highly specialised to target well-behaved properties (order parameters) of systems and adapted to take advantage of systems near-equilibrium or in equilibrium for the sake of efficiency [21–25]. This limits their applicability when attempting dynamical self-assembly design, where it is important to assess time-dependent thermodynamical and dynamical effects. The multi-scale assembly mechanisms, from the short-time annealing of local defects to the long-time cluster merging events, introduce kinetic features, e.g. defects and cluster morphology, to the assembling structures that may cause high variability between independent realisations of the self-assembly, producing large fluctuations in the assembly outcome. Cluster morphologies are of concern for many functional structures [3; 16]. They can have knock on effects in hierarchical assembly where the progressive assembly of intermediate structures is used to bridge the gap between the small colloids and large targeted structures [26; 27]. With the emergent self-assembly outcome of systems highly sensitive to the initial conditions and particle design, direct particle simulations of systems remain an integral part of the modelling of these dynamical systems. The more inefficient but reliable forward design thus remains important for tuning design parameters of dynamical systems.

The goal of the work in this thesis is to speed up and reduce the cost of searching for optimal interaction parameters in systems where direct particle simulation—which alone are already computationally intensive and require long execution times—is the most reliable way to model and evaluate the particle system of interest. An examination is performed on the stochastic hill-climbing algorithm, a well-known and easy-to-implement local search heuristic, as an alternative to the simple brute-force scanning of the parameter space for the tuning problem. Interacting externally with the particle simulation program, the hill-climbing algorithm iteratively informs the simulation inputs as it seeks higher yielding solutions, effectively "climbing" the yield landscape. In this sense it is an attempt to reduce the computational demand and execution time of the forward design approach when finding the interaction parameters that maximise the self-assembly yield. We also empirically explore a selection of metaheuristic strategies and tools, applicable to the hill-climbing algorithm with noisy measurements, to seek ways to shorten the search intensity and execution length and time. These metaheuristic strategies may also be applicable to the improvement in efficiency of inverse design schemes where the iterative algorithm for converging to the desired target solution is based on a local search algorithm that assesses

its current solution at every iteration, and the evaluation of particle systems remains expensive.

To this end, an example parameter tuning problem is considered: we study the tuning of interaction parameters for a dynamical self-assembling system of short-ranged attractive patchy particles to optimise its target yield outcome, based on the structural quality and morphology of the largest cluster in this system. We define a *cluster* to be an energetically-bound collection of disks interacting with at least one other member via a patch-patch interaction. For this, a patchy hard disk model system is chosen as the model colloidal system of interest, providing two interaction parameters that form the design (parameter) space of the parameter tuning problem, with each state point on this space representing a solution to the tuning problem. The patchy particle model is important as a surprisingly effective coarse grain model for polymers, applicable especially to biological systems, and is seen as one of the most basic non-trivial anisotropic colloidal systems, forming the bridge between theory and experiment, and a blueprint for more complex and ambitious colloidal designs. The self-assembly of the particle systems are modelled using the Virtual-Move Monte Carlo (VMMC) algorithm by Whitelam et al. [28]. This algorithm, based on moving clusters of particles in unison, is chosen to provide more realistic dynamics for mimicking the physical collective behaviour of a real physical system that are lacking in standard Monte Carlo modelling approach involving single particle moves.

Initialised from out-of-equilibrium states, direct particle simulations are executed to measure the performance of the emergent assembled structures under different interaction parameters values. The parameter space is systematically scanned to provide a benchmark for the parameter tuning problem. The yield outcomes of these systems form a yield landscape on the parameter space with the tallest peak corresponding to the sought parameters maximising the target structure assembly. Crucially, each self-assembly run of the system will yield a different measurement, resulting in a noisy yield landscape.

Our focus then switches to parameter tuning using the stochastic hill-climbing algorithm, synonymous with finding this highest peak—the global maximum. The stochastic hill-climbing algorithm is assessed empirically on a model version of the noisy yield landscape. This noisy yield landscape model is constructed based on the simulated data to substitute the execution of time- and resource- expensive particle simulations to enable a large sample of hill-climbing runs to be generated within reasonable timescales. Several metaheuristic tools and search strategies to complement and augment the stochastic hill-climbing algorithm are also considered to improve the search outcomes at termination while attempting to reduce the computational demand. In particular, a trial period limit m is considered to limit the number of evaluations of the algorithm's current solution, providing an "investment-free" parameter to control the noise felt by a climber. Moreover, the cumulative averaging of yield measurements over several evaluations of a solution is considered as opposed to the conventional method of averaging over a larger number of simulations per evaluation.

The objective of the hill-climbing (meta-)heuristic is to provide an alternative method to the conventional brute-force scan, offering options in search method and flexibility in the tailoring of the hill-climbing. This aims to extend the tools available to users looking to

tune parameters, whose choice of search method will ultimately be based on a combination of search objectives (what is required from the search) and the practical constraints in execution. In fact, the generality of our assumptions makes this relevant not only for users performing any intensive simulation based sampling with an optimisation problem based on searching a 2D parameter space, but also for any users adopting the hill-climbing algorithm who are concerned with the computational performance of the heuristic.

1.1 Thesis Outline

This thesis is set out as follows. In Chapter 2, an introduction to colloidal systems is given, highlighting their potential to organise into interesting structures. This is followed by a description of the core concepts of self-assembly. Targeted design and parameter tuning in the context of self-assembly are reviewed, focusing on how one can search for higher yielding targeted assembly more efficiently.

Chapter 3 describes the fundamentals of particle simulation using Monte Carlo techniques and the VMMC algorithm used to simulate the assembly of the patchy hard disk model system, which is defined in Chapter 4.

Chapter 4 reviews the existing knowledge of self-assembly of the patchy disk model, before moving onto our own self-assembly simulations of the model. We make the choice to target the self-assembly of large, round, compact honeycomb clusters. To capture the target assembly quantitatively, we constructed a yield measure consisting of two parts: a measure of the quality of honeycomb crystalline domains formed, and a shape factor measuring the closeness of the resulting cluster to a compact circular aggregate. The observations of the self-assembly performance across the parameter space are described, and from the simulations a yield landscape of the system is obtained.

A more mathematical framework for optimisation and search heuristics in the form of finding local and global optima on a fitness landscape is introduced in Chapter 5. In particular, a review of local search heuristics on noisy fitness landscapes, relevant to our parameter tuning problem with noisy yield measurements, is presented, considering techniques used to handle the fluctuations in measurements.

This mathematical interpretation is then used to formalise the study of the self-assembly parameter tuning problem. In Chapter 6, the yield landscape toy model is constructed on which the performance of several variants of the stochastic hill-climbing algorithm, obtained by combining different protocols to form a complete algorithm, is assessed. After elucidating the effects of the trial limit period m on the hill-climbing behaviour, the optimal m value and strategies exploiting the nature of the self-assembly yield landscape are presented.

Chapter 2

Colloidal self-assembly and particle design

Self-assembly is the phenomenon describing how components organize themselves relative to each other to form ordered structures and patterns, without any direct control provided by external sources. This definition can be viewed in a very broad context, for example in the field of engineering and robotics where simple programmable units obey set rules to self-assemble into larger structures by analyzing both individual and collective behaviour [29]. We concern ourselves with self-assembly at the molecular and meso- scales. The collective assembly behaviour of particles is a result of the components trying to minimise their free energy whilst obeying the second law of thermodynamics.

Self-assembly is a fundamental aspect of biological processes, where it plays a vital role in the construction of complex macromolecular assemblies found in organisms. Examples include the formation of the double helical DNA through hydrogen bonding of the individual strands, and the folding of proteins and interactions with other polymer groups, where erroneous arrangements can have devastating consequences [30–32]. Research in supramolecular chemistry and nanotechnology also utilises self-assembly for the synthesis of microscopic compounds and structures [8; 33]. With the ability to get components to assemble or disassemble under specific conditions, applications include catalysts [3] and drug delivery [6; 7].

We focus our attention on colloidal self-assembly that shows the ability to form ordered structures. An important focus of current research is understanding how the relative local interactions between such microscopic components govern the collective behaviour and assembly pathways leading to a target structure, and to ultimately manipulate the particle interactions to embed the desired self-assembly process into its design. Reproducibility, timescales, and stability are crucial to predicting whether a thermodynamically designed system can reach its intended final state. Computational modelling can help determine assembly pathways and mechanisms, as most intermediate states are transient and hard to detect in experiments.

In this chapter, we will introduce colloidal self-assembly, then review some approaches to manipulating structures via local anisotropic interactions that demonstrate how self-assembly mechanisms of patchy particles have been exploited to produce novel structures.

We follow this with a summary of the current progress in targeted design of particle interactions and parameter tuning.

2.1 Colloidal systems

Colloidal systems consist of microscopic particles and a fluid medium to hold them. These colloids, roughly 1-1000nm in size, are small enough that thermal motion of the solvent molecules are important, resulting in a diffusive random motion of the colloids. The colloids are thus allowed to explore possible configurations as the system settles into an energetically favourable state. Particles that are larger than the microscale do not experience a net displacement from the weak random thermal forces of the solvent; stronger forces are required to move them around [17]. On the other hand, colloidal particles are of the size where thermal fluctuations give rise to diffusive (Brownian) dynamics, allowing them to explore different configurations as they seek to relax into gas, liquid or solid phases, in a manner similar to molecular systems relaxing under equilibrium thermodynamics. For larger colloids, this occurs on time scales that allow observations to be made under a microscope, providing a good system for scientists to study statistical mechanics and thermodynamical behaviour. Additionally, colloidal particles tend to interact weakly with each other, allowing entropic contributions to play an important role in determining the behaviour of the colloidal system.

Classically, the study of crystallization and how matter organise itself has been restricted to isotropic particles, borrowing from well-studied theories of nucleation and phase behaviour to understand the basis of assembly. A limited number of different crystals have been obtained using spherical particles with isotropic potentials in monodisperse systems, with polydisperse (particles with varying mass) systems an outlet for modifying the phase behaviour of a system [34]. Anisotropy, however, is present in most systems of interest; organic compounds, proteins and simple molecules are not isotropic in shape or interactions. A consequence of this anisotropy is an increased specificity and complexity of bond arrangements in addition to lower densities of ordered structures [35]. New phases are possible even with the simplest of changes in colloidal geometry [36]. The diversity of self-assembled structures only continues to increase as the variations in particle design expand. Moving from isotropic particles to anisotropic ones is hence desirable when searching for new structures and functionality.

This motivation is coupled with the advancement of increasingly sophisticated and intricate fabrication of non-trivial colloidal particles. Possible colloidal features to modify include, but are not limited to, geometry (rods [36], dumbbells [5], polyhedra [19; 37], topological knots and links [18; 38]), composition (polydisperse [9; 10; 36], mixtures [4]), and varying surface interactions (patch valency [20; 39; 40], patch surface roughness [41; 42], indented [43]).

Colloidal systems provide an alternative way of producing functional materials. Synthesis of such materials focus on the building blocks rather than the final structure, so scaling up its manufacturing is easier than a process based on building the large scale framework before working on the finer details. Controlled assembly/disassembly of mis-

celles and reconfigurable structures from shape-shifting colloids [44–46] provide promising mechanisms for novel applications in smart materials and for biomedical purposes [47].

For larger colloids in the range of optical length scales, colloidal crystals can be used for their photonic properties. Photonic crystals consisting of anisotropic particles show promise in creating more controllable photonic materials. For example, colloidal dumbbells were successfully assembled into a 3D crystal with tunable optical properties [5]. With no field, the dumbbells are able to form a plastic crystal—a positionally ordered crystal with rotational freedom. However, by applying an electric field, a uniform orientation is achieved, enabling optical properties in colour and variable refractive index (birefringence) to emerge.

2.2 Patchy colloidal particles

Patchy colloidal particles refer to colloids that have been manipulated to produce directional interactions. Their surfaces can be decorated with chemically-attractive patches to restrict the ways neighbouring particles ‘stick’ to each other, or molded in non-uniform ways to produce geometric features such as depressions, creating differing interactions on its surface. Patchy particle systems have been found to exhibit rich phase behaviour [27; 48–52]. The break in orientational symmetry of the units have significant consequences on the resulting self-assembled structure, favouring certain orders that reduce its free energy under the prescribed environment to access a range of possible self-assembled structures otherwise inaccessible through isotropic interactions. Variations in surface position and distribution, number and size, and a combination of spots, stripes and rings can give rise to both regular lattices and pseudocrystals, amorphous solids, as well as unexpected phase behaviour such as plastic crystals [53]. Aside from enthalpic forces, entropic forces have also been shown to be important in the ordering of systems, especially hard particles. This stems from depletion forces as the efficient ordering of colloidal particles free up available volume in the system for the solvent (depletant) to explore that would otherwise be lost (an inefficient packing would create an abundance of volume inaccessible to the depletant). Entropic patches created via geometric modifications have been considered to access an even richer colloidal design space [9; 10; 35].

More advanced methods of fabricating patchy particles with rich topology and variability have been reported recently. Stellacci and coworkers showed that a range of tunable patchy particles can be constructed by applying a binary mixture of immiscible alkanethiol surfactants (different end-groups to the alkane chain) of different lengths to coat spherical nanoparticles [54]. Varying the ratio of the binary mixture changed the domains of microphase separation of the two mixtures, creating intricate stripes and spotted surfaces on the coated nanoparticles. Pons-Siepermann and Glotzer extended this direction of research by looking at ternary [55] and quaternary [56] mixtures. Several design rules were identified using simulation, based on the relative ratios of surfactant lengths and nanoparticle size. These mixtures settled into configurations that reduced the contacts between immiscible mixtures to minimise energy, while encouraging interfaces between surfactants of different length or bulkiness to maximise conformational entropy, producing

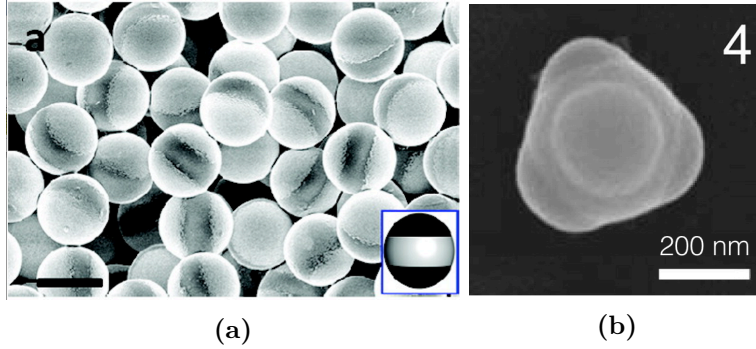


Figure 2.1: (a) Patchy colloidal particles with metal coating (black in inset). Reprinted with permission from [40]. Copyright 2011 American Chemical Society. (b) Tetrahedral patchy particle. Reprinted with permission from [41]. Copyright 2009 American Chemical Society.

a rich spectrum of possible surface patterns. These studies showcase possible routes to intricate designs of patchy particles that can be used for self-assembly of complex structures. Targeting more richly patterned building blocks is a gateway for creating even more diverse structures.

Short-ranged hard-core patchy particles have received huge attention as candidate patchy models. Under suitable conditions, short-ranged hard-core patchy spheres have been shown to prefer liquid states over solid states at zero temperatures [50]. Here, the tetrahedral spacing of four patches on a sphere directly anchors the bonding directions of the neighbours required for diamond structure, which is favoured at the correct density for finite temperatures. However, widening the patches allows the stable liquid phase to extend towards zero temperatures at certain packing fractions. These patches can only share one bond at any one moment and thus when a patch is in contact with two other patches at the same time, one of two possible bonds can form, multiplicatively increasing the number of possible configurations with the same particle positions. Extended to the full system, for packing fractions where the average number of contacts per patch can be maximised in a disordered state, the increased configurational entropy stabilises this liquid phase at zero temperatures.

From the design and fabrication of patchy colloidal particles with hydrophobic caps at opposite poles, Chen et al. were able to observe the self-assembly of such colloids (similar to the patchy coverage found in Fig. 2.1a), suspended between solutions of different densities, into the targeted two-dimensional Kagome lattice under suitable conditions [39]. Experimental work like this on self-assembly of patchy colloidal particles not only confirms the approach to studying the thermodynamically stable structures, but also provide valuable information on the dynamical self-assembly pathways chosen by the system. Subsequent simulation work based on the same model [49; 57] showed that this open crystalline arrangement owes its stability to rotational and vibrational fluctuations. These modes increase the elastic free energy of the system compared to a closed packed arrangement—the preferred configuration for hard core and short-ranged isotropically interacting spheres.

Such entropic effects have been responsible for observed preference of assembly structures in many patchy particle systems where enthalpically similar configurations are com-

peting [13; 50]. The extra room for particles to move out of their lattice sites and rotate without breaking bonds help stabilise configurations. This ability to create open structures is something not offered by geometry driven self-assembly [11]. For extended attractive patches, the extra room between interacting particles introduces vibrations. An analytical explanation shows that this vibrational entropy also helps stabilise open lattices.

Further control in the final product can be introduced by incorporating different species of patches to favour specific patch-patch bonds, promoting/restricting the permitted configurations both locally and globally. Moreover, different interaction pairs may bond at different rates, enabling control of hierarchical assembly pathways on a multitude of scales [58; 59], or to add chirality to structures [27]. At the other extreme, a full multi-component design of structure consisting of many unique species of building blocks has been shown to produce very specific structures [59]. DNA can offer such highly specific interactions [60]. Contrary to the theory that systems with growing specificity and detail take longer to assemble, fully multicomponent assembly has been shown to give reliable self-assembly characteristics at reasonable timescales, both experimentally [61] and via simulation work [62; 63]. With the misbinding of DNA pairs energetically disfavoured, target structures are designed to be the unique energetically favoured structure. Using them to decorate colloids so that every component is distinct allows one to create assemblies of detailed finite structures [64].

The range and diversity in techniques, colloidal design and self-assembled structures introduced in this section demonstrate the vast possibilities in particle design and potential diversity in assembled structures and system of interest. Computational modelling and direct simulation remains a cost-effective approach to discovering, searching and predicting good target self-assembly design and behaviour. We finish this section by referring the reader to Duguet et al. [48] that gives an excellent review on the current state of research on patchy particles.

2.3 Core concepts of colloidal self-assembly

Thermodynamics indicates the favourability of possible states, with a closed system tending to realise a global free energy minimum. Colloidal self-assembly considers the processes as a disordered system of colloidal particles evolves into an ordered state. As particles move in the solvent, interacting with each other, enthalpic and entropic contributions are minimised and maximised respectively, lowering the free energy of the system in the process. As pair interactions are typically weak (e.g. volume depletion, van der Waals, hydrogen bonding), particles can sample several different configurations until stabilised by surrounding particles collectively.

Bond specificity is often required for well-ordered structures. For example, one could attach attractive patches at the specific tetrahedral points on spherical particles to promote the assembly of (double) diamond structures [50; 65; 66]. Patches with a small solid angle result in a rigid and precise diamond structure once assembled. Strong interactions help stabilize the structures once formed.

However, as particles swim about interacting with each other, it is inevitable that they

will bond in non-ideal ways which, if the bonds are too strong, will persist and disrupt the local order of a structure, inhibiting the growth of the equilibrium structure. Non-specific bonds lead to many local bonding mistakes incommensurate with the desired assembled structure. The inability for defects to dynamically anneal out during self-assembly is known as *kinetic trapping*. An abundant level of kinetic trapping disrupts good assembly, leading to disordered aggregates. Therefore effective assembly requires the ability to anneal defects naturally.

To suppress kinetic trapping, the specificity of the bonds can be increased to reduce the number of possible configurations, and thus the number of incorrect bonding, allowed [67]. However, patches still need to be wide enough. Interactions that are too directional require precise alignment to form bonds, lowering the chances of patches of finding each other in the first place.

Similarly, the bond strength ought to be weak enough to allow any undesired bonds to break and test out several other arrangements to find those commensurate with the ordered structure, but strong enough to stabilise the structure. Good assembly therefore occurs in a finite window of parameter values, sandwiched between the absence of assembly and poor quality aggregation [28] (for example see Fig 4.16a). It is believed that good assembly requires reversible interactions, where the rates of bond-forming and bond-breaking are of a similar order of magnitude [68]. Microscopic reversibility is seen as a crucial property to ensure that bonds can be made and broken easily during assembly to allow defects to annealing out, producing an ordered final state [68–71]. This highlights the delicate balance of the self-assembly process to achieve significant and rapid growth that is also high in quality. In a physical system this typically involves weak interactions such as hydrogen bonding and van der Waals.

Microscopically-reversible interactions also coincide with the argument suggesting that a localised region of small patches leads to a more efficient assembly than a large patch of identical interaction strength. More specifically, the nucleation and growth of a system of particles with n small weakly-interacting patches each with strength U_i and distributed in a localised region R interaction exhibits higher activities of bond breaking and bond forming compared with a similar system of particles with these regions R substituted by a single patch with interaction strength $n \sum U_i$ [72]. The multi-level interaction energies accessed by collective pair interactions of the former means that the effective interparticle interaction is more reversible, thus misbinding is not as costly as the latter case when a bond does form.

The combination of thermodynamical preference and kinetic factors thus determines whether a system will relax into the preferred equilibrium state or become arrested in some metastable state, and the *self-assembly pathway* it takes to get there. Particles can directly agglomerate into a stable structure, or transition through several intermediate metastable states [67]. The pathways followed by a system depend on the initial state of the self-assembly, which determines whether the system starts near-equilibrium and is dominated by reversible interactions, or starts from a far-from-equilibrium state. Kinetic trapping arises when particles bond too quickly without unbinding fast enough, often resulting in metastable states inconsistent with desired ordered structure.

Self-assembly of isotropic spherical particles have been well explored previously in terms of their predictability, preferred equilibrium state, and growth behaviour. This is epitomised by Noro and Frenkel’s work extending the law of corresponding states (all simple gases obey the same reduced equation of state when scaled by their thermodynamic variables e.g. temperature and volume) to colloidal suspension of spherical particles with attractive pair potentials [73]. On the other hand, anisotropic colloidal systems remain relatively unexplored, due to the complexity introduced by even the simplest of changes to the particles, and the diversity of variations. New structures can be constructed that are otherwise not possible from using isotropic spheres only. Efforts to study anisotropically interacting, anisometric particles [74] have been made.

Aside from studying the behaviour of these new colloidal designs experimentally, simulations of these colloidal models have provided us with accurate observations of self-assembling systems, enabling the exploration of their dynamical and thermodynamical phase spaces. Simulations of self-assembly systems are typically equilibrium studies or dynamical experiments. The ability to track intermediate states permits the study of competing assembly mechanisms underlying the emergent collective behaviour to numerically observe the self-assembly pathways. The ease in varying model parameters makes computational studies a useful tool for the tuning of self-assembly designs.

2.4 Designing self-assembling systems

While studying the equilibrium phase behaviour is key to knowing whether a colloidal system will eventually self-assemble into a target structure, it is often necessary to know if the correct self-assembly will occur under a given initial state within experimental and practical times. The exploration of a system’s *dynamical phase behaviour* is thus crucial for identifying regions of parameter space in which a target structure is emergent. Moreover, one can target kinetically trapped states that are only examinable through dynamical studies, or maximise the yield of a given target structure by controlling the level of self-assembly activity, or lack thereof.

There are an increasing number of studies and reviews distilling the mechanisms introduced by colloidal features that are predicted to encourage self-assembly of particular types of structure [9; 37; 48; 75; 76]. These *design rules* provide valuable guidelines on what to incorporate into the specifications for a more fruitful design. Equally as important, these rules inform us about the colloidal features to avoid, reducing the parameter space of the problem.

Dynamical events or modifications to the environment influence assembly pathways to produce alternative structures. External factors such as surface substrates [77; 78] and seed structures [65] alter the free energy landscape of the system and lowering nucleation barriers to encourage certain assembly pathways that would otherwise be unfavourable. The assembly of capsids, such as nanoparticles and viral protein capsids around RNA, can be enhanced by introducing alternative assembly mechanisms via templating [79]. Shape-shifting particles or time-dependent protocols offer time-dependent assembly pathways, allowing ordered assembly to occur for a simpler geometry before morphing into

different shapes [46; 80], or to adjust nucleation and growth rates in a system of competing structures [63; 81]. Hierarchical assembly also offers alternative scalable pathways [26; 27; 59; 82; 83]. Examples like these offer additional kinetic factors that extend the assembly pathways of the system. Accurate modelling of these dynamical systems require intensive computer simulations.

2.4.1 Targeted design

For a more purpose-driven design process of self-assembly systems, it is important to start with the goal in mind. Desired configurations and topological features of structures are captured by a set of specifications with which the design outcome aims to satisfy. Informed selection strategies of colloidal features using the design rules introduce different assembly mechanisms into the system. Interaction parameters require further tuning to bring out a correct balance in assembly mechanisms to yield the desired emergent structures. There are broadly two approaches to designing and tuning colloidal particles, thought to be fundamentally different in philosophy: forward design and inverse design.

Forward design

The conventional *forward approach* or direct approach involves applying design rules to combine intuitively sound interaction features together with a handful of free parameters and predict the structure of the aggregate based on knowledge of the interaction potential. Computationally these interaction parameters are tuned by performing an exhaustive scan (brute-force scan) of the whole parameter space, by taking a grid of parameter values at a set resolution and simulating the respective systems, in order to identify the values maximising the yield.

The major downfall of such strategy is the inefficiency of the process, both in design and in parameter tuning. Our interpretation of the design rules limits the complexity and creativity of the design. More non-trivial target structures usually require more intricate features beyond the straight-forward blending of basic design rules. The brute-force approach to parameter tuning is inefficient, and scales poorly with the number of parameters considered. However, forward design remains an integral approach due to its flexibility in application. The search process is independent of the system modelled, which can be very simple, in the case of direct self-assembly in a closed system, or more complex, such as time-dependent interactions and hierarchical assembly.

Inverse design

Inverse design starts with the desired attributes in mind and evaluates the candidate model in comparison to the target. Candidates yielding structures that better satisfy design criteria are deemed fitter. The performance of the model is redirected as input to modify the candidate design, typically with a large number of design variables to consider, until the optimal solution is found. The idea is to avoid a full exploration of the configuration space of the system and the design space of the model involved, while allowing the algorithm to

access non-trivial designs and explore systems that cannot be easily inferred using basic design rules.

These highly specialised optimisation algorithms, or heuristics, have mostly been developed to look at in-equilibrium or near-equilibrium systems, where target structures can be characterised by time-independent distributions or well-behaved order parameters with low fluctuations. They iteratively generate and test solutions, discarding non-optimal ones in favour of better performing parameter values, until the ‘best’ design is found. Genetic algorithms (GA) and evolutionary algorithms (EA), aside from being used to identify favourable energy minimising particle configurations (see Section 2.4.4), are used to evolve an initial set of base colloidal designs to maximise the similarity between candidate and desired self-assembled structures, e.g. DNA-grafted colloids [24].

A target configuration can be characterised using a radial distribution function (RDF) $g_{tgt}(r)$ that shows positional correlations between particle pairs of distance r apart. Starting with some functional form, a piecewise pair interaction potential with a large number of parameter variables is tuned according to $g_{tgt}(r)$. For example, Jadrich, Lindquist and coworkers [21–23] applied iterative schemes to optimise pair potentials of finite range to target non-trivial microstructures (e.g. cluster fluids, porous mesophase) using the differences between the current iteration’s RDF $g_i(r)$ and the target RDF $g_{tgt}(r)$ to adjust the interaction potential, known as Iterative Boltzmann Inverse (IBI). This technique can be performed using multiple target data to tune models whilst satisfying multiple objectives e.g. anisotropic systems characterised by orientationally-dependent correlation functions [84] or a system that is consistent over a range of thermodynamic states [85]. The concept is general enough to be applied to other parametric functions, such as the tuning of size distributions in polydisperse systems [86].

Recent advances saw a data-driven approach to parameter tuning for realising particular cluster configurations of double-ring colloids [25]. Using a combination of various techniques involving particle simulation, diffusion mapping and hybrid Monte-Carlo sampling of this diffusion space, both the thermodynamical and dynamical effects of the system were accounted for in order to converge onto parameters that yielded and maximised the self-assembly of octohedral clusters from a well-mixed initial fluid state. This marks progress in inverse design incorporating dynamical behaviour, adding to techniques such as Ref. [24] that folds in the kinetics of self-assembly into the design and prediction of crystalline structures using GA primed with a library of predicted assembly mechanisms and models. Data-driven approaches take advantage of the computational power to work with multiple sets of data simultaneously.

With multiple degrees of freedom in the system considered, coupled with numerous design parameters to modify, the vast design space demands huge computational resources. An increase in system size or number of free interaction parameters to tune leads to a polynomial increase in computational effort. This currently limits the practicality of inverse heuristics to studying small crystalline systems, time-dependent systems for short durations, equilibrium systems, and disordered states (e.g. fluid clusters). These systems restrict the size of available configurations and/or limit the timescales involved in order, reducing the complexity of the structure and allowing simulations to exploring the system

more effectively to obtain larger statistics (and hence more accurate measurements). However, many desired systems and practical applications of structures require self-assembly of systems involving larger length scales or longer timescales. In these cases, direct simulation of the particle system remains a reliable approach, able to incorporate the dynamical details as well as keeping the thermodynamics accurate.

Choosing between the two approaches

As we have illustrated, inverse design has received great attention in recent years due to its advanced capabilities in interpreting and modifying complex multidimensional parameters. This is seen as a crucial component for accessing a richer design space and the subsequent novel structures. Indeed, the ability to utilise characteristics of the desired target system configuration allows for a more efficient and purposeful strategy in computer-assisted targeted design. However, although the inverse design philosophy is straightforward, these heuristic approaches so far address parameters with particular qualities. More creative target criteria are subject to the suitability of the available inverse techniques (as a single algorithm or a combination of different techniques), thus more novel systems may not have the adequate conditions for these advanced techniques to be implemented.

The forward approach is an ever present methodology for exploring such systems at the forefront of development. Its simplicity offers a foundational strategy for exploring uncharted territory in design space and novel systems. Moreover, its independence from the modelling and simulation techniques of the system of interest makes it a ready-to-use strategy during any point of a project, although brute-force scanning, the only obvious method in this category, is not efficient. We explore whether hill-climbing search is able to offer a more resource efficient technique in the context of parameter tuning and exploration.

2.4.2 Parameter tuning

Parameter tuning plays an integral role throughout the design process. Design components brought together using design rules need tuning to enable emergent target assembly. Once the design components have been decided, further tuning is required to maximise the desired assembly performance. It is often desirable to maximise the efficiency, and therefore yield, of the self-assembly to lower production costs. The scope of this thesis deals with the tuning process after design components have been decided. This can be treated as an optimisation problem, where the optimal solution is the set of design parameters values satisfying the target specification. This is revisited in Chapter 5. Of the many possible design parameters available for tuning for hard particles (e.g. aspect ratio, patch positions) we focus on the tuning of the interaction strength of attractive patches and its coverage on the surface of hard disks.

Inverse methods utilise parameter tuning for targeting both dynamical and structural qualities of the system. For example, Klotz and Jack [87; 88] developed a feedback mechanism to tune the interaction strength of short-ranged attractive hard spheres on-the-fly, such that the self-assembly system exhibited reversible dynamics. Reversible dynamics (i.e. equilibrium in time [28; 67–70]) is believed to be a good indicator of well-behaved self-

assembly with relatively few defects. Nguyen et al. [89] used the particle radial distance from its parent cluster as the control parameter to renormalise the repulsive interactions relative to the attractive ones to form a uniform-sized aggregation of both polydisperse and monodisperse particles into supraparticles. The absence of this renormalisation results in gel-like structures percolating the system.

Parameter tuning in the forward approach is motivated by the physically observable self-assembly outcomes of the system, often a single valued quantity. Many studies explore the self-assembly behaviour on a two-dimensional parameter space by evaluating a set of parameter points (or *state points* or *solutions*) that covers the entire space. Typically, a two-dimensional (2D) parameter space is explored by evaluating a finite set of state points uniformly distributed on a grid, upon which the interaction parameters are varied [72; 75; 90; 91]. A brute-force scan is performed where the respective system is evaluated at each point via simulation, and the point corresponding to the best yield outcome is identified as the optimal solution given the grid resolution. The additional bonus of a brute-force scan is the resulting dynamical phase diagram. Further tuning at a finer resolution and/or smaller region of parameter space is possible to increase the accuracy of the solution.

The yield of the system on the parameter space can be thought of as a landscape, with the ‘height’ of a point a measure of its performance (fitness). Yield measures are often defined such that the resulting landscape peaks at the desired target. Locating the interaction parameters that maximises the yield of the system is therefore the same as finding the global peak (maximum) of the yield landscape. The reduction in yield value as one moves away from the optimal parameters reflects the deviation from assembly of the target structure. Multi-staged assembly schemes require the tuning of multiple yield landscapes [26]. It is this exploration of parameter space in the forward approach that we focus our attention on in the rest of this thesis.

2.4.3 Speeding up the exploration and optimisation of parameter values

Given a bounded parameter space deemed interesting and a chosen simulation method to sample the self-assembly result for a particle system, it is desirable to maximise the time spent obtaining information at the region of interest, i.e. the region of parameter space that gives significant yield. In the brute-force scan mentioned in the previous section, each region is sampled in proportion to its fractional area. Regions of good assembly are typically small, occupying only a fraction of parameter space; most of the time and resource is spent simulating regions of low yield, making it highly inefficient for searching high yielding parameter points. One can reduce the demand of the scan by increasing the grid spacing, but at the cost of lowering the resolution of the landscape.

A move away from uniform brute force scanning would reduce the time and cost of exploration. Search heuristics, a large topic of study in mathematics and computer science, have been applied to solve many numerical problems. Instead of sampling the full problem space, search methods use data obtained to iteratively guide the exploration of space, so that an answer is reached by only covering a subset of points in the problem space. We cover the topic of heuristic search in more detail in Chapter 5.

We found only one study that addresses heuristic exploration and search of parameter space in the context of self-assembly landscapes. Phillips applied adaptive mesh refinement (AMR) to iteratively distribute evaluations in a parameter space based on previous evaluations to discover all distinct regions (i.e. phases) present in the parameter space [92]. The algorithm coarsely covered the entire parameter space with a mesh of points, before iteratively inserting more points to sample about small and/or interesting regions, along boundaries between regions, and/or new regions of unexplored space, redistributing the time and resources spent for exploring different regions of space evenly. This method does not consider noise in the measurement, although one can reduce the noise by averaging over multiple independent samples of the same state point to obtain an accurate measurement.

The measurement of the self-assembly yield is intrinsically noisy. Yield measures that are based on locally average order parameters (e.g. bond-orientational order) have small variance in noise. Other parameters can vary wildly between different realisations (e.g. cluster shape). It is common practice to reduce a state point evaluation by averaging over several yield measurements. Unlike in-equilibrium studies where several observations averaged over time is encouraged, in dynamical studies the observation is time sensitive and an average is obtained by taking the mean over individual simulations. Increasing the number of simulations inevitably increases the computational cost. The use of advanced computational tools such as shape matching and machine learning [93] to assist the detection and characterisation of structures can bring insight to the development of new structural measures for handling more complex assembly products.

In summary, the current strategy of brute-force scanning of parameter space is highly inefficient. On top of this, reliable identification of the high yielding state points requires repeated measurements to reduce the effect of noise, leading to a high computational cost. This motivates our study of local search algorithms and strategies to aid search on a noisy landscape and reduce the overall computational cost involved by cutting down on the number of simulations called.

2.4.4 Speeding up particle simulations

Long and intensive computer simulations are required to simulate extended periods of self-assembly of large and/or complex systems. In practice, an important component in reducing the execution time and computational resources spent in parameter tuning is the better modelling and more effective simulating of each individual particle systems. Indeed, much effort has been devoted to improving the accuracy and efficiency of these simulation models. In this section we highlight some of the recent work addressing these computational challenges, mainly for Monte Carlo (MC) simulations.

There is no time dependence in equilibrated systems hence particle dynamics are not crucial to understanding the globally favourable state of the system. This gives freedom in the choice of update moves in an MC algorithm to increase the sampling efficiency. Unphysical non-local moves, such as cluster flipping in geometric cluster algorithm [94; 95] or long distance moves in basic Monte Carlo, help overcome poor sampling in locally-dense

configurations

Biased MC methods [96–98] are another popular approach used to traverse phase space more efficiently and promote the sampling of low probability regions of phase space. A biasing term is added to the Boltzmann contribution in the move acceptance criterion of an MC algorithm, transforming the underlying probability distribution sampled by the algorithm. If $\text{acc}(\mu \rightarrow \nu) \propto \exp(-\beta H)$ is the acceptance criterion for moving from state μ to state ν with a Hamiltonian energy difference ΔH (See Section 3.2) then, with weight factors w_i associated with each state, this acceptance is modified into $\text{acc}(\mu \rightarrow \nu) \propto \exp(\beta \Delta H) \cdot \exp(w_\mu - w_\nu)$ ($\beta = 1/k_B T$ with k_B the Boltzmann constant and T temperature). By choosing to modify the Boltzmann factor by the inverse likelihood of observing the state, exploration of likelier regions of phase space are suppressed in favour of sampling low probability regions of phase space. This is used to overcoming large free energy barriers to from one (meta)stable state to another and spend more time sampling regions of interest [97].

Genetic algorithms and evolutionary algorithms are used to identify favoured configurations that minimise the enthalpy for zero temperature ($T = 0$) systems [52; 66; 99]. Starting from a set of candidate system configurations, each configuration is examined and those with the lowest energy are used to generate the next set of candidate configurations. Bianchi et al. [100] showed that Monte Carlo and evolutionary algorithms gives consistent results on crystalline phases of small patchy particle systems, but with the Monte Carlo approach offering more flexibility by being able to handle finite T and discontinuous interaction potentials.

Dynamical self-assembly studies ask *how* and *if* the system will reach its thermodynamically preferred location in phase space. One may also target dynamically arrested states such as amorphous crystals [13]. For this, kinetic features such as binding and unbinding events and internal relaxations are important to the system and hence modelling of the time-dependent dynamics are necessary. Large system sizes, slow assembly mechanisms and rare-events are among the reasons for the long execution times required.

Techniques for improving the speed and efficiency of dynamical MC simulations avoid decorrelating the states generated between successive iterations of the MC algorithms. In diffusion-driven self-assembly, the maximum step sizes of moves are restricted, but cluster moves model correlated motion to improve acceptance rates [101]. Forward flux sampling tracks and promotes relevant but rare events, pushing simulations to spend more time sampling low frequency events to generate better statistics [102; 103]. Kinetic Monte Carlo focuses on the coarse grain modelling of the dynamics, incorporating faster modes into slower ones to restrict the move pool and advance the system at larger time steps [104]. This is effective for structure growth and diffusion studies where the important kinetic processes and transition rates are identifiable and can be used as inputs for the Kinetic Monte Carlo algorithm. Simulations with implicit-solvent and/or coarse grain modelling offer alternatives to explicit modelling, providing efficient and effective modelling of self-assembly at longer characteristic times [105; 106].

2.5 Conclusion

Practical applications of self-assembly studies for targeted design requires the consideration of thermodynamical and dynamical understanding of the system. Once interaction parameters have been identified, it is important to find out how such parameters couple together to produce the desired structure, and at which parameter values the resulting colloids can optimally assemble. Direct simulations of large systems remain a reliable approach for tuning interaction parameters at latter stages of targeted design, where systems are highly sensitive to input parameters. A brute-force scan of the parameter space is not ideal in the parameter tuning problem as simulations and experiments cost execution time and computational resources. In this thesis, we study the application of hill-climbing algorithms and variants for finding optimal or near-optimal solutions in parameter space, based on comparing the intrinsically noisy yield performance between solutions. We empirically confirm whether this approach is practical and introduce a few metaheuristics and strategies for improving this approach.

Chapter 3

Monte Carlo simulations

Numerical simulations form an integral part in the study of self-assembling systems. They form the bridge between theory and experiment, testing theories by applying them and resolving the physics behind experimental results, as well as predicting phenomena that are hard to observe experimentally. Computer simulations also offer consistency in controlled and reproducible testing, and can track the system's evolution down to particle level and individual interactions. For designing self-assembly systems, computational modelling is used to discover potential designs and parameter values before they are tested physically.

We use numerical simulations to simulate our model colloidal system. Specifically we use Monte Carlo (MC) techniques to obtain particle trajectories relevant to different particle systems under a simple system. MC is based on generating states of a system based on the equilibrium distribution of states without explicitly considering the particle dynamics.

In this chapter we review Monte Carlo methods as an application of statistical mechanics and thermodynamics used to explore particle systems, and how realistic dynamics can be approximated in the limit of small update moves. For a more comprehensive description of statistical mechanics and Monte Carlo methods, refer to reference material such as Newman and Barkema [107], and Frenkel and Smit [98]. We then introduce the virtual-move Monte Carlo (VMMC) algorithm, a member of the MC family designed to move a collection of particles concurrently in order to improve the sampling of collective dynamics. Note that in this chapter, the term cluster refers to a general collection of particles labelled as a single entity due to some criteria (e.g. when the energetic evaluations are non-zero), and not limited to those with patch-patch interactions as defined specifically for our patchy hard disk system (See Section 4.1).

3.1 Modelling particle systems

There are several distinct classes of numerical techniques for modelling particle systems, each with their own advantages and disadvantages. Among them Molecular dynamics (MD) and Monte Carlo (MC) methods are the most popular. Molecular Dynamics models the movement of molecular components in a system to generate configurations through solving Newton's equations of motion for an N body system. By explicitly considering

particle motion, energy is exchanged between molecules via collisions, enabling free energy barrier crossings which lead to better sampling of configurations in denser systems, e.g. in liquid systems. On the other hand, Monte Carlo relies on generating a random configuration for each move based on space of available states in order to derive the equilibrium average of some property A . Proposal moves are accepted with a likelihood based on the energy of the system, reflecting the likelihood of the state itself. Choosing which technique to use is largely determined by the problem being investigated and the information sought from the simulations. MD is favoured to analyse dynamics, seeing heavy use in the study of biological systems where explicit modelling of all components in the complex system in a time sensitive manner is paramount. On the other hand, MC is popular for equilibrium studies, with the versatility of update moves able to efficiently explore large numbers of states to obtain better statistics. For example, there is a large probability of selecting random states that result in overlapping molecules in dense systems. A large number of rejected moves leads to poor efficiency in sampling. However, random moves trigger barrier crossing events more readily in low density systems. In comparison, MD can struggle due to the lack of collisions, so molecules become stuck sampling a fraction of low energy configurations.

MC methods get around the problem of inefficient sampling by performing unphysical moves, such as flipping particles/molecules around, and moving particles in non-local ways, including particle addition/deletion in Grand Canonical Monte Carlo where the particle number in the system varies. Additionally, the lack of force calculations mean larger systems and longer timescales can be accessed at relatively cheaper computational costs. MC can model a large range of ensembles. Transport properties such as diffusion coefficients are naturally derivable from the equations of motion in MD but lacking in MC.

3.2 Monte Carlo method

Monte Carlo sampling methods generate random states proportional to their likelihood under the equilibrium distribution of the system's phase space, subject to the chosen statistical ensemble. We will focus on the canonical ensemble, where the number of particles N , the volume V , and the temperature T of a closed system are constant. A sequence of states generated based solely on the current state is 'memoryless' (it does not depend on any previous states) and is known as a *Markov chain*. This builds up a simulation pathway that samples the system of interest in an ergodic manner, yet one that is more efficient than considering all possibilities available in the system.

Metropolis-Hasting Monte Carlo avoids the evaluation of the partition function Z . Instead, a proposal distribution is conditioned on the current state. Typically, a small random translation or rotation is applied to a single particle in an existing configuration to generate proposal moves. This new state is accepted with a probability proportional to the ratio of importance weights of the states before and after the move. This conditioning of the acceptance probability pushes the simulation to sample more favourable regions more often. This is known as *importance sampling*. The rate of sampling is therefore built from the underlying probability distribution, rewarding more favourable states by visiting

associated configurations more often than less favourable ones.

3.2.1 Statistical Ensemble

Consider a d -dimensional system of N anisotropically interacting particles in a box of volume V at temperature T . Each particle has position $\mathbf{r} = (x_1, x_2, \dots, x_d, \theta_1, \theta_2, \dots, \theta_d)$ and momentum $\mathbf{q} = (p_1, p_2, \dots, p_d, L_1, L_2, \dots, L_d)$, comprised of spatial x_i and orientational θ_i positional components, and linear p_i and angular L_i momenta components, respectively. The full set of position and momentum coordinates can be used to define a microstate of the system. For a system in thermal equilibrium with a reservoir at temperature T , the probability density of a state with \mathbf{r}^N and \mathbf{q}^N is related to the Boltzmann factor corresponding to the total energy H of that configuration,

$$P(\mathbf{r}^N, \mathbf{q}^N) = \frac{e^{-\beta H(\mathbf{r}^N, \mathbf{q}^N)}}{Z}, \quad (3.1)$$

where $\beta = 1/(k_B T)$, k_B is the Boltzmann constant, and Z is the partition function

$$Z(\mathbf{r}^N, \mathbf{q}^N) = \int e^{-\beta H(\mathbf{r}^N, \mathbf{q}^N)} d\Gamma, \quad (3.2)$$

representing all points $\Gamma(\mathbf{r}^N, \mathbf{q}^N)$ in the phase space Γ , specifically for a constant NVT system.

When an observable quantity of interest A of the system is independent of the particle motion, we can consider the integral to be dependent on \mathbf{r}^N only. The independent \mathbf{r}^N and \mathbf{q}^N coordinates allow for the factorisation of Z into a sum of kinetic and potential parts, or equivalently, the Hamiltonian as a sum of the potential U and kinetic K energies $H(\mathbf{r}^N, \mathbf{q}^N) = U(\mathbf{r}^N) + K(\mathbf{q}^N)$.

Assuming classical particles of mass m and moment of inertia I , there is quadratic dependence of the kinetic terms on momenta components (linear kinetic energy $p_i^2/2m$ for each p_i , and rotational kinetic energy $L_i^2/2I$ for each q_i). Using the relationship $\int_{-\infty}^{\infty} \exp(-u^2) du = \sqrt{\pi}$, these kinetic terms can be analytically integrated to constants and hence can be factorised out of Z .

Equation (3.1) then becomes

$$Z(\mathbf{r}^N) = \int e^{-\beta U(\mathbf{r}^N)} d\mathbf{r}^N \quad (3.3)$$

with probability of a state occurring being

$$P(\mathbf{r}^N) = \frac{e^{-\beta U(\mathbf{r}^N)}}{Z}. \quad (3.4)$$

To numerically sample this partition function we generate the states according to the underlying transition rates and probabilities that would be observed for the physical system. However, instead of producing a full integration of all possible states in the ensemble, feasible only for the smallest and simplest of systems, the focus here is to obtain a representative sample of states proportional to the likelihood of them occurring. A

system in a state μ will evolve into state ν after time t with some rate characterised by a transition probability $\pi(\mu \rightarrow \nu)$. We utilise this set of transition probabilities to transform the system in order to explore the different possible states. The generation of these state need to be random, so that even from the same initial state the random sequence of samples generated by the algorithm is different every time. To do this, we use a *Markov process* – a random process satisfying two conditions: (1) the transition probabilities are time-independent; (2) the transition probabilities depend solely on the properties of the current states μ and ν and no other states. Such processes are set up such that if it is run for sufficient time, from any initial state, it will eventually produce a sequence of states that appear with probability given by the Boltzmann distribution, analogous to a physical system reaching equilibrium. Two additional conditions are sufficient to guarantee that the system converges towards this underlying probability density distribution, regardless of where we start looking in the neighbourhood of the phase space, and is trajectory independent [107]: ergodicity and detailed balance (see below).

The resulting sequence of states generated using a Markov process is known as a Markov chain. One can then use this Markov chain to query observables of the system. In-equilibrium studies are usually concerned with the expected outcome, or ensemble average, for some observable quantity $A(\mathbf{r}^N)$ of the system obtained by performing a weighted average of A with respect to all possible states in a system to give $\langle A \rangle$:

$$\langle A \rangle = \frac{\int d\mathbf{r}^N A(\mathbf{r}^N) e^{-\beta U(\mathbf{r}^N)}}{\int d\mathbf{r}^N e^{-\beta U(\mathbf{r}^N)}} \quad (3.5)$$

$$= \int d\mathbf{r}^N A(\mathbf{r}^N) P(\mathbf{r}^N). \quad (3.6)$$

The integration over all states is replaced by a weighted averaging of A with respect to the sample of states generated by the Markov process. When the number of points sampled M is sufficiently large, the number of points m_i generated around \mathbf{r}^N reflects the probability ratio, so that

$$\langle A \rangle \approx \frac{1}{M} \sum_{i=1}^L m_i A(\mathbf{r}_i^N). \quad (3.7)$$

3.2.2 Ergodicity and detailed balance

To ensure that a Markov Chain Monte Carlo algorithm will produce a sample distribution that reflects the underlying equilibrium probability distribution given enough time, it needs to satisfy *ergodicity* and *detailed balance*. Applied to particle systems, the former guarantees that all states can be reached from any other, and the latter ensures that the sampled states and transition probabilities collectively reflect the Boltzmann distribution [107].

Given a dynamical system, ergodicity is the quality of the transformation of the system where the average behaviour of the system over time is equivalent to the statistical average of all the states visited by the system. This property allows us to compute a measurable quantity by either time averaging or ensemble averaging as the system evolves. More crucially, it is necessary in order to guarantee that all possible states of the system can

be explored given enough time, regardless of the starting configuration, hence no state is prevented from being explored by construction. An ergodic Markov chain is one where any state can be reached in at least a finite number of steps from any other state.

Physically, detailed balance is a principle that came to represent the concept of microscopic reversibility. Time reversibility for elementary collisions representing kinetic processes, once broken down into steps, implies that at equilibrium, any collision is equilibrated by an equal but opposite collision. For Markov processes, detailed balance can be interpreted as a condition where the average number of accepted moves from state μ to state ν is exactly the same as the reverse transition from ν to μ . This can be written as

$$P(\mu)\pi(\mu \rightarrow \nu) = P(\nu)\pi(\nu \rightarrow \mu), \quad (3.8)$$

where $\pi(\mu \rightarrow \nu)$ is the transition matrix relating the probability of a system in state μ moving to state ν . This is a sufficient but not necessary condition [108] that guarantees the satisfaction of a broader relation, *balance*, given by

$$\sum_{\nu} P(\nu)\pi(\nu \rightarrow \mu) = \sum_{\nu} P(\mu)\pi(\mu \rightarrow \nu) = P(\mu), \quad (3.9)$$

with the left equality describing how the rate of transitions in and out of any state μ must be equal at equilibrium, and the right equality from applying the definition that Markov processes must generate some valid state ν when given any state μ , i.e. $\sum_{\nu} \pi(\mu \rightarrow \nu) = 1$. Equation (3.8) guarantees the satisfaction of the left equality in Equation (3.9), which is a necessary condition that implies $P(\mu)$ will be the equilibrium probability of the Markov process. In addition to this, it can be shown that systems satisfying the condition of detailed balance will always converge to this equilibrium distribution from any state, something that the left relation in Equation (3.9) by itself doesn't guarantee [107; 109].

From Equation (3.8) we select the Boltzmann distribution to be the equilibrium distribution that the Markov process will converge towards:

$$\frac{\pi(\mu \rightarrow \nu)}{\pi(\nu \rightarrow \mu)} = \frac{P(\nu)}{P(\mu)} = e^{-\beta(U(\nu)-U(\mu))}. \quad (3.10)$$

The transition matrix $\pi(\mu \rightarrow \nu)$ can be expressed as the product of two independent parts:

$$\pi(\mu \rightarrow \nu) = \alpha(\mu \rightarrow \nu)\text{acc}(\mu \rightarrow \nu), \quad (3.11)$$

the underlying Markov transition matrix $\alpha(\mu \rightarrow \nu)$ that determines the probability of state μ transitioning to state ν , and $\text{acc}(\mu \rightarrow \nu)$ which provides the probability that the proposed move $\alpha(\mu \rightarrow \nu)$ is successful.

Substituting Equation (3.11) into Equation (3.10), we get

$$\frac{\alpha(\mu \rightarrow \nu)\text{acc}(\mu \rightarrow \nu)}{\alpha(\nu \rightarrow \mu)\text{acc}(\nu \rightarrow \mu)} = e^{-\beta(U(\nu)-U(\mu))}. \quad (3.12)$$

As long as this ratio is satisfied, the method for generating new states from old states and

the details of the acceptance criteria can be tailored to suit the system being studied [107].

A stricter subset of this condition is superdetailed balance [98], used when a probabilistic artificial labelling of the particles in state μ is performed to give a realisation \mathcal{R} . Here, the probability of generating the same labelling \mathcal{R} in both μ and ν is explicitly considered in the balance equation:

$$P(\mu)\alpha(\mu \rightarrow \nu|\mathcal{R})\text{acc}(\mu \rightarrow \nu|\mathcal{R}) = P(\nu)\alpha(\nu \rightarrow \mu|\mathcal{R})\text{acc}(\nu \rightarrow \mu|\mathcal{R}). \quad (3.13)$$

This is used in cluster algorithms where each distinct way of grouping disks into a single entity is a different realisation.

3.2.3 Metropolis Monte Carlo

A basic MC sampling scheme can now be constructed. As mentioned previously, there is freedom in choosing the state generator and acceptance probability to sample the phase space of a system as long as Equation (3.12) is satisfied. In general this choice is done to maximise the sampling efficiency [107]. The widely used Metropolis scheme is the standard approach for generating states. Let a model system be in state μ with positions \mathbf{x}^N and orientations $\boldsymbol{\theta}^N$. Random translation and rotation moves altering the available coordinates of the system have a maximum magnitude of Δ_t and Δ_r respectively. One scheme that proposes moves of a single particle at a time is as follows:

1. Calculate the energy of the system in current state μ , $U(\mu) = U(\mathbf{x}^N, \boldsymbol{\theta}^N)$.
2. Select a particle with coordinates $\mathbf{x}, \boldsymbol{\theta}$ at random, and choose to displace the particle with probability p_t (Go to step 3), otherwise rotate the particle i.e. the probability of performing a rotation is $p_r = 1 - p_t$ (Go to step 4).
3. Translate the particle by a random displacement vector δ_t with uniformly chosen step size in $[-\Delta_t, \Delta_t]$, i.e. $\mathbf{x}' = \mathbf{x} + \delta_t$, $\boldsymbol{\theta}' = \boldsymbol{\theta}$. Go to step 5.
4. Rotate the particle a random rotation vector δ_r with uniformly chosen step size in $[-\Delta_r, \Delta_r]$, i.e. $\boldsymbol{\theta}' = \boldsymbol{\theta} + \delta_r$, $\mathbf{x}' = \mathbf{x}$. Go to step 5.
5. Calculate the new energy of the system in state ν , $U(\nu) = U(\mathbf{x}'^N, \boldsymbol{\theta}'^N)$, and accept this move with probability

$$\text{acc}(\mu \rightarrow \nu) = \min \{1, \exp(-\beta[U(\nu) - U(\mu)])\}. \quad (3.14)$$

6. End of a move. Go to step 1.

This Metropolis scheme assumes a symmetric Markov transition matrix (i.e. $\alpha(\mu \rightarrow \nu) = \alpha(\nu \rightarrow \mu)$), with zero probability for transitioning to states that are not possible. The acceptance scheme in step 5 is the defining part of the Metropolis approach. This form maximises the probability of accepting any moves, while satisfying the acceptance ratio between $\mu \rightarrow \nu$ and $\nu \rightarrow \mu$ transitions. For in-equilibrium studies, maximum step sizes Δ_t and Δ_r are typically chosen to increase the acceptance rate of proposed moves to

maximise the simulation efficiency. They can also be chosen to satisfy other characteristic rates, such as the diffusion rates for simulating dynamical system.

3.2.4 Simulating dynamical systems

When out-of-equilibrium systems are studied, one is typically interested in how the system evolves from an initial state. The sequence of states generated by MC methods naturally defines a trajectory associated with a time axis t . Measurements of the evolving system are therefore also time dependent i.e. $A(t)$. Einstein’s relations can be used to obtain a physical timescale for diffusive dynamics of the system, at which point the relation between Δ_t and Δ_r can be established.

Colloids move under Brownian dynamics, diffusing in the solvent in order to interact with each other. Brownian motion can be approximated by overdamped equations of motion, where the average acceleration is zero due to the viscous effects of the solvent. One can apply equations of motion by Langevin [110] to model such diffusive systems, typically using Brownian Dynamics methods [111] or Molecular Dynamics simulations [112]. However, it has been shown that the Monte Carlo approach can be adapted to evolve a system under an approximate Brownian dynamics regime, being applied to many dynamical studies. Overdamped Langevin dynamics can be approximated using MC by restricting update moves to local physical moves (translations and rotations) limited to small step sizes [113; 114]. Although dynamic details and forces are not considered explicitly to generate moves, the resulting motion from Monte Carlo moves may reflect an integration of overdamped Langevin equations of motion. An accurate approximation is possible by relating the diffusive timescales with acceptance rates and move update sizes, and is applicable to multicomponent and anisotropic systems [115–117].

For our self-assembling system, we considered a canonical ensemble of 2D patchy hard disk system in an implicit solvent initialised in a well-mixed fluid condition, and advanced this system using Metropolis Monte Carlo updates to relax the system. We initially used a simple Monte Carlo algorithm for our simulations that proposes single particle moves. However, the acceptance rate was low due to the large energy differences involved with the highly directional and short-ranged potentials, and the high percentage of proposed moves in locally dense regions that led to the forbidden overlap between hard disks. Moreover, the strongly interacting disks showed a lack of collective particle motion, resulting in very unrealistic movement. To counter this, we adopted the *Virtual-Move Monte Carlo* (VMMC) algorithm [28; 101] that explicitly considers moving groups of particles to restore collective dynamics in the system. The use of VMMC led to a significant improvement in the self-assembly yield outcomes of our low density patchy particle system, with a much higher particle agglomeration activity and more realistic dynamics.

3.3 Virtual-Move Monte Carlo (VMMC)

The virtual-move Monte Carlo (VMMC) algorithm [28] is a probabilistic state-generating algorithm that considers moving groups of particles collectively. VMMC was invented

to restore the collective dynamics lacking in Single Particle Monte Carlo (SPMC), when translations and rotations are restricted to local moves (small step sizes). By doing so, one can retrieve an approximate overdamped Langevin dynamics. A pseudocluster is generated by selecting a random seed particle, proposing a move from the move pool, before scouting for nearby particles interacting with the seed particle that will be affected. The recruitment of neighbouring particles, known as *linking*, is subject to probabilities correlated with cluster size. The resulting pseudocluster is then treated as a single unit and moved under an appropriate acceptance criterion. The acceptance criterion considers relative diffusion rates, scaling with the size of the cluster (assumed to be spherical/circular), which are crucial to studying systems where the main modes of clusters growth are not via the agglomeration of single particles.

Collective motion is suppressed in SPMC; a set of particles has to move sequentially over a series of complementary moves in order to mimic collective motion. Reducing the update step size can restore the collective behaviour of particles. However, this comes at the cost of advancing the system at finer time steps, which requires more iterations of the algorithm and is especially inefficient at low densities where particles need to travel far to interact with other particles. Moreover, this approximation is poor in some conditions such as when particle-particle interactions vary strongly with distance and angle, as large energy differences lead to low acceptance rates of such systems, compromising the evolution of the system [118]. To avoid this slowdown yet regain the long range collective motion, explicit consideration of collective particle (cluster) moves is built into the Monte Carlo sampling. Such cluster moves allow direct sampling of these collective modes, enabling long ranged correlated dynamics to be realised explicitly in the simulation trajectories while avoiding further reduction in the update step sizes. We use the VMMC algorithm to alleviate this problem for the patchy particle system.

Explicitly referenced in its name, VMMC performs ‘virtual’ moves on bodies in the system, whether it is an individual particle or a group of particles. Proposed Monte Carlo moves consider both the initial state before a move is applied, and the final state after it is applied. This amounts to considering the energy change, or energy gradient, associated with the move for each particle affected [118]. The diffusion of the moving particles is also incorporated into the move proposal, in order to impose a relation between cluster size and diffusion coefficient. The algorithm assumes the moving group to be spherical (circular in 2D). The virtual move is then subject to the acceptance condition, and is implemented as a proper move if accepted.

3.3.1 Move proposal

A particle in the box, a proposed move type (translation/rotation) and the move details (direction and magnitude) are all randomly selected at the start of a VMMC move. The particle becomes the seed particle of the virtual move. For translation moves, proposed with probability p_t , the seed particle is selected to move locally within a radial distance Δ_t . For rotation moves, attempted with probability $p_r = 1 - p_t$, the seed particle is rotated about its disk centre by a randomly chosen angle in the range of $[-\Delta_r, \Delta_r]$ around

a randomly oriented axis \hat{n} . In a 2D system, \hat{n} is perpendicular to the plane of the box.

Once a virtual move is chosen, neighbouring particles are iteratively recruited into the moving particle group, or *pseudocluster* \mathcal{C} , by testing the link between neighbouring pairs before and after the virtual move. Beginning in a microstate μ , a particle j outside of the moving cluster lying next to a member i of the moving group \mathcal{C} are linked with a probability

$$p_{ij}(\mu \rightarrow \nu) = \Theta(n_c - n_{\mathcal{C}}) I_{ij} \max(0, 1 - e^{\beta(\epsilon_{ij} - \epsilon_{i'j})}), \quad (3.15)$$

where ϵ_{ij} is the interaction energy between the particle pair i and j , and $\epsilon_{i'j}$ is the energy between them after the virtual move is applied to i into its new position as i' in the resulting microstate ν .

The factor $\Theta(n_c - n_{\mathcal{C}})$ ensures that motions of clusters of different sizes are not oversampled so that all particles experience proposed moves with approximately equal frequency. A random number ι drawn uniformly from the interval $[0, 1]$ is chosen before the move to give a maximum cluster size cutoff $n_c = \lceil \iota^{-1} \rceil$. The move is rejected if the number of particles $n_{\mathcal{C}}$ in the moving pseudocluster grows larger than n_c , ensuring that clusters of all size are moved with the same frequency. I_{ij} is a boolean matrix identifying particle neighbour pairs i and j .

The recruitment process finishes when no more particles are waiting to be tested. The resulting move of the pseudocluster \mathcal{C} , defining the transition between states μ to ν , is then subject to the move acceptance criterion. The axis of rotation is fixed at the COM of the seed particle.

3.3.2 Move acceptance

Using the above linking scheme results in a move $\mu \rightarrow \nu$ being accepted with probability

$$W_{acc}(\mu \rightarrow \nu) = \Theta(n_c - n_{\mathcal{C}}) D(\mathcal{C}) \times \min \left\{ 1, \prod_{\langle i,j \rangle_{n \leftrightarrow o}} e^{-\beta(\epsilon_{i'j} - \epsilon_{ij})} \prod_{\text{failed}} \frac{q_{ij}(\nu \rightarrow \mu)}{q_{ij}(\mu \rightarrow \nu)} \prod_{\text{linked}}^{\mathcal{R}} \frac{p_{ij}(\nu \rightarrow \mu)}{p_{ij}(\mu \rightarrow \nu)} \right\} \quad (3.16)$$

satisfying super-detailed balance (Section 3.2.2). $\langle ij \rangle_{n \leftrightarrow o}$ denotes particle pairs that start off interacting in state μ and non-interacting in ν , or start off non-interacting in μ but ends up interacting in ν , i.e. particle pairs that see a change in energy. $q_{ij} = 1 - p_{ij}$ is the probability of links not forming, i.e. failed links, internal and external to \mathcal{C} , and \mathcal{R} denotes a particular realisation of formed and failed links built during the recruitment procedure.

$D(\mathcal{C})$ is a diffusion cutoff parameter that enables one to enforce dynamics of collective motion at different scales. In the limit of a tightly bound cluster that is also assumed to be isotropic, pseudoclusters are always entire clusters and Stokes' Law can be implemented with $D(\mathcal{C})$ chosen to reduce the rate of accepting a move according to the cluster size. Smaller clusters diffuse at a larger rate than a larger particle according to

$$D_t = \frac{k_B T}{6\pi\eta R_H}, \quad D_r = \frac{k_B T}{8\pi\eta R_H^3}. \quad (3.17)$$

k_B is the Boltzmann constant, T is the temperature, and R_H is the hydrodynamic radius of the cluster. The diffusion coefficients D_t and D_r scale inversely with R_H and R_H^3 for translation and rotations respectively. To enforce this, a random number γ from a uniform distribution on the interval $[0, 1]$ is chosen and a translational move is terminated immediately upon n_c exceeding R_H/γ (R_H/γ^3 for rotational moves). This is accurate for pseudoclusters that are spherical (disk-like in 2D). Further attention is required for the parameterisation of clusters away from this limit, with the consideration of relative translation and rotation move proposal probability p_t and p_r and step sizes Δ_t and Δ_r required to include approximations for internal relaxations of a less tightly bound cluster, and anisotropic clusters. A possible treatment to increase the accuracy of moves of anisotropic \mathcal{C} is to numerically resolve the average diffusive properties of different cluster shapes and sizes, as demonstrated in Ref. [90]. For our purposes, we did not perform such parameterisation to obtain an accurate dynamics, instead borrowing the VMMC ability to move groups of clusters efficiently to allow the quick spatial localisation of colloidal particles from an initially dispersed state.

We used the VMMC library developed and maintained by Hedges [119] that implements a symmetric version of this algorithm [28; 101]. In this symmetrized version of the algorithm, a proposed move is applied on a seed particle in its initial microstate twice, once in the intended ‘forward’ direction, and once in the ‘reverse’ direction. Failed links in the first ‘forward’ stage are not considered in the second stage—all other particle pair are considered in the second stage. Failed links from the second stage are considered as frustrated links—they result in an inconsistency in the realisations considered by the forward and inverse moves. Only particle pairs that form links successfully in both directions are considered as fully linked, and these fully linked particles form the moving pseudocluster.

This symmetrized recruitment scheme leads to a simplified acceptance probability procedure, exploiting the fact that only failed links external to the cluster contribute to the acceptance rate [28]. By construction, the observed failed links internal to the moving cluster, and all the fully formed links are the same pre- and post- move. These failed links and fully formed links therefore cancel each other out in the multiplicative sums of the link probabilities in Equation (3.16), reducing the acceptance probability to a simpler form that only requires the consideration of particles pairs contributing to $\prod_{\langle i,j \rangle_{n \leftrightarrow o}} e^{-\beta(\varepsilon_{i'j} - \varepsilon_{ij})}$, i.e. either those that go from interacting to non-interacting, or not interacting to overlapping. Frustrated links joining the pseudocluster leads to immediate rejection of the move [28].

3.4 Conclusion

The Metropolis MC and VMMC algorithms for modelling particle systems were introduced in this chapter. Although the assumptions for the algorithm are based on sampling the phase space characterised by the underlying equilibrium probability distribution, a dynamical interpretation of the system’s evolution is possible by restricting the step sizes of the updates.

The VMMC algorithm, a powerful cluster algorithm, was adopted to improve the dynamics observed in the simulations, as strongly interacting hard disks were inefficiently

evolved using SPMC. VMMC enables cluster moves which collectively update an interacting group of particles based on the consequences of applying the move. The patchy hard disk system in solution exhibited more realistic activity in solution after adopting the VMMC algorithm, with cluster-cluster agglomeration and internal relaxation events producing aggregates that were not realised under the SPMC regime at much longer times. As we will see in the next chapter this has huge consequences on the final cluster shape.

Chapter 4

Self-assembly of a patchy disk system

The 2D patchy particle model studied by Whitlam et al. [13] is an example of a non-trivial self-assembly system that exhibits a wide range of different structures when interaction parameters are varied, able to form a variety of ordered and non-crystalline structures depending on the interaction strength and the flexibility of the patches. The patchy model is popular as a coarse grain model for systems with directional valency, such as polymer systems and globular proteins, where anisotropy of subunits is an important factor governing the complexity of aggregate morphologies. The self-assembly and growth of protein structures is pertinent to the understanding of deficient assembly [32], a cause of diseases such as cataracts [31], Parkinson’s and Alzheimer’s. The 2D patchy disk model we used is based on the widely used Kern-Frenkel model [49; 74; 120–122]. Despite the surprisingly complex and rich phenomena expressed by this model, its theoretical and computational simplicity make it a popular choice in computer modelling of polymer chain and ring formation. We adopt this model as our example system to study the targeted design of high yielding self-assembled structures, by searching for the design parameters that maximise the yield of large round compact honeycomb clusters.

In this chapter, we introduce the patchy particle model and summarise the previous findings by Whitlam et al. on such a system. Next, a discussion of yield measures characterising the degree of the target structure formation is given. Measuring only the largest cluster in the system, our target structure is a symmetric, round, compact, large honeycomb cluster. We capture the network quality and domain size using the bond-orientational susceptibility, and the cluster morphology (specifically the compactness and circularity) by constructing a shape factor, such that products more similar to the target assembly product give higher values. By quantifying and combining these features into a single yield measure Q , a yield landscape defined on the parameter space is constructed, with the best yielding interaction parameters at the global peak of the landscape. The VMMC self-assembly simulation setup is described, followed by a presentation and discussion of the simulation results and the self-assembly behaviour of the patchy disk model systems.

4.1 Patchy disk model

The 2D patchy particle model consists of hard disks of radius $a/2$. Each disk is decorated with three uniformly distributed patches—each labelled with a bisecting patch vector \mathbf{u}_γ —on the surface sweeping an angle of $2w$ from the disk centre, forming sectors. These sectors are arranged uniformly around the disk such that their bisector \mathbf{u}_γ are at an angle $2\pi/3$ relative to each other. Only in the neighbourhood perpendicular to the patched surfaces (the arc of the sectors) does a square well potential exist, extending from the particle surface at radius $a/2$ to a radius of $(a + \delta)/2$, where $\delta = 0.1a$. A patch of *interaction strength* ε has a well depth of $-\varepsilon$.

Disk pairs i and j interact via a patch-patch interaction if the displacement vector \mathbf{r}_{ij} between the centres of i and j intersect a pair of patch arcs, one from each disk, and no longer than $a + \delta$ (See Fig. 4.1). To sum up, the potential of the patchy model is

$$V_p(\mathbf{r}_{ij}, \Theta_i, \Theta_j) = \begin{cases} \infty & \text{if } |\mathbf{r}_{ij}| < a, \\ -\varepsilon V_{ang}(\mathbf{r}_{ij}, \Theta_i, \Theta_j) & \text{if } a \leq |\mathbf{r}_{ij}| \leq a + \delta, \\ 0 & \text{if } |\mathbf{r}_{ij}| > a + \delta, \end{cases} \quad (4.1)$$

where \mathbf{r}_{ij} is the displacement between two disk centres, and Θ_i is the orientation of particle i , and

$$V_{ang}(\mathbf{r}_{ij}, \Theta_i, \Theta_j) = \begin{cases} 1 & \text{if } \theta_{\gamma ij} < w \text{ and } \theta_{\gamma' ji} < w, \\ 0 & \text{if otherwise,} \end{cases} \quad (4.2)$$

where $\theta_{\gamma ij}$ is the angle between patch perpendicular bisector \mathbf{u}_γ on particle i and the interparticle vector \mathbf{r}_{ij} , as shown in Figure 4.1. We limit w , the *flexibility* of a patch, to a maximum of $\arcsin(\frac{a/2}{a+\delta}) = \arcsin(5/11) \approx 27^\circ$, such that each patch may only interact with at most one other patch from a different disk at any one time. w is called the flexibility as it represents the freedom of the relative position between a patch-patch interacting pair while remaining bonded. We impose this upper bound on w to avoid contact between patches belonging to three different disks. Otherwise, additional rules are required to determine which of the equally valid particle pairs interact, which can lead to interesting behaviour [50].

As we shall see in the next section, a diverse phase behaviour can be produced by varying only the interaction strength ε and flexibility w . A two-dimensional model is also easy to represent, inspect and characterise via a yield measure, hence we choose this as our example system.

4.1.1 Review on this particular model

As mentioned previously, this colloidal model is taken from the study performed by Whitlam et al. [13]. In their work, the bulk behaviour of the system is explored in the grand-canonical regime, characterising the competing factors in interaction strength ε and patch flexibility w that result in a rich thermodynamical and dynamical phase behaviour (see

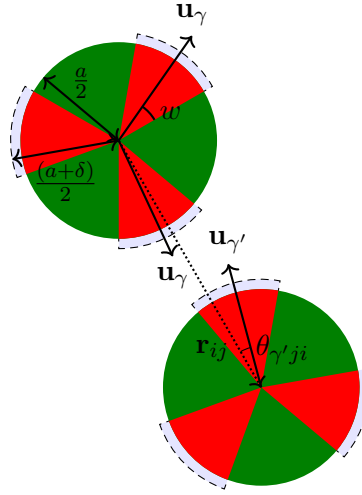


Figure 4.1: Schematic of patchy disk model of radius $a/2$. Three sectors (red) indicate the angle subtended by the surface arc where attractive patch-patch potentials (blue regions) interact with energy $-\varepsilon$. Patches γ on disk i and γ' on disk j interact if the displacement vector \mathbf{r}_{ij} between the disk centres is no longer than $a + \delta$ and passes through γ and γ' .

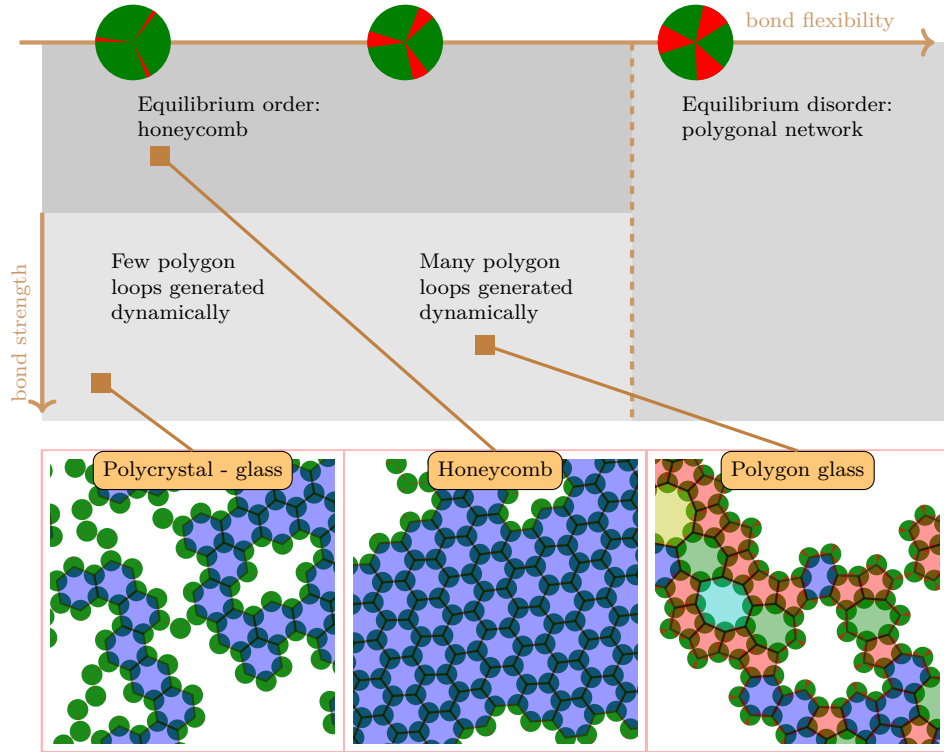


Figure 4.2: Schematic phase diagram of the patchy hard disk model as a function of the interaction strength ε and flexibility w , annotated with example systems. At equilibrium, networks are found to be honeycomb at low w or polygonal at high w . When the interactions are stronger, kinetic effects play a more significant role, with early-formed polygon networks annealing into honeycomb networks. When interactions are very strong (high ε), kinetic defects persist, leading to dynamical self-assembly networks such as polycrystals in low w and polygon glasses at high w . Figure adapted from [13].

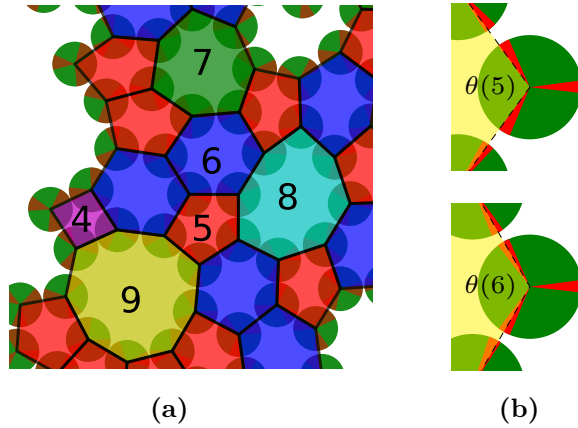


Figure 4.3: (a) Key legend for polygon loops of different sizes, determined by the number of disks composing the loop. Polygon loops are filled according to their size: pentagons are red; hexagons are blue; and heptagons are green. (b) Demonstration of the restricted movement of a disk in non hexagonal polygons. A disk in a pentagon loop has less freedom to rotate (top), compared to a disk in a hexagonal loop (bottom).

Figure 4.2 adapted from Ref. [13]). The resulting self-assembled 2D networks resemble various real physical systems at different scale magnitudes with topologically distinct features, achieved by only resolving these two interaction parameters. For example, cyclohexanem-phenylene assembly [78] was identified to be a honeycomb polycrystal in a regime that resembles the assembly of equilibrium honeycomb with dynamically induced grain boundaries due to inflexible (low w) yet strong (high ε) bonds. With a single simple model able to reproduce the assembly behaviour of several real physical systems of a variety of length scales and assembly components, Whitelam argued that this strongly suggests that basic ‘design criteria’ (in this case geometry and strength of binding) play a dominant role in determining the stable structures preferred by systems, both thermodynamically and dynamically.

The diverse range of emergent networks stems from the local interactions between neighbouring particles. As the patches bind to each other, the patchy disks each create up to three pairwise bonds to form an open network structure, with the closed convex loops of neighbouring disks resembling polygons (Fig. 4.3a). Disks distributed in a regular hexagonal manner form ordered, periodic *honeycomb networks*; disks organised in a less ordered configuration through a mixture of non-hexagonal loops and incomplete loops form a non-periodic *polygonal network* (or *polygon network*). Six patchy disks can bond to form a hexagon loop, the preferred loop size complimenting the three-fold symmetry of the three-patch design; the 120° angle between patch sectors within a disk matches the internal angle of a regular hexagon. Hexagonal loops form naturally for any $w > 0^\circ$, and rotate freely by up to $2w$ without breaking bonds (Fig. 4.3b bottom). Patches need to be wide enough to form other polygonal loops, as well as having smaller rotational freedom once formed. For example, a regular pentagon loop has an interior angle of 108° , so can only be accessed by patchy disks with $w \gtrsim 6^\circ$ (Fig. 4.3b top).

At equilibrium, the 2D patchy disk model is predicted to favour a honeycomb network at small w , but prefer a polygon network at large w (bottom centre in Fig. 4.2). For very

low w , disks are only able to access hexagonal loops. Considering isolated polygon loops, when w is large enough for other polygon loops to form, the hexagonal configuration is still thermodynamically favoured as it maximises the disks' rotational entropy compared to other polygon loops. Carried over to a system of patchy particles, this entropic contribution leads to the preference of honeycomb lattice in a fully-connected network, especially for small w .

As the free energy difference between different isolated polygon loops reduces with increasing w [13], other polygon loops become more stable relative to the hexagon case. Polygonal networks are found to be favoured at large w since the available distinct network configurations that particles can arrange into increases in number; the rotational entropy contributions from the disks are outweighed by a larger configurational entropy of available network configurations. This was argued using analytical estimates from studying a mean-field topological-gas [13]. The irregular distribution of disk neighbours enables the resulting polygon loops to tessellate 2D space, stabilising these polygonal defects.

Dynamically, even more network types are generated upon varying ε and w . A periodic honeycomb lattice is still the preferred network structure at low w . Grain boundaries are low in numbers for weak bonding disks, but permeate throughout the system in the strong bond (high ε) regime, creating polycrystals (bottom right of Fig. 4.2). For more flexible interactions (high w), polygon glass form (bottom left of Fig. 4.2). In fact, polygon loops, especially pentagons, form readily when patches are wide enough, but are unstable. If interactions are weak enough then ample time is available for any polygon loops, a source of kinetic trapping, to anneal out into hexagons. Strong interactions leave existing loops persisting for long times, leaving the network in a glassy state with kinetic defects.

Although our system will be at a low constant density, we expect our simulations to exhibit similar dynamical self-assembly behaviour seen here, with the effects of varying interaction strength and flexibility, dictating the dominant assembly regime and kinetic trapping, enough to reproduce the various networks seen here.

4.2 Simulation details

Our simulations model the dynamical self-assembly of patchy hard disks from an initially far-from-equilibrium, disordered well-mixed fluid state. A system of $N = 900$ particles, initiated from a disordered state in a square container of area fraction $\varphi = 0.04$ with periodic boundary conditions, is relaxed using the VMMC algorithm for 2×10^7 VMMC cycles. One VMMC cycle is equivalent to performing N VMMC move attempts. Translation and rotation moves are attempted with equal probability, with constant maximum step sizes of $\Delta_t = 0.13a$ and $\Delta_r = 10^\circ \approx 0.175$ respectively. These values were taken from previous equilibrium studies and do not result in a consistent time scale δ_t according to the Einstein relations ($\Delta_t = \sqrt{6D_t \times \delta_t}$ and $\Delta_r = \sqrt{2D_r \times \delta_t}$). Only one of these step sizes is required as independent variable if the time scale δ_t and the relations are respected. The low area fraction provides adequate room for clusters to form while avoiding percolation, where a cluster extends beyond the box dimensions and interacts with a periodic copy of itself, forming gels. We take ε to be the scaled interaction strength i.e. $\varepsilon = U/T$ for an

interaction of energy magnitude U at temperature T . Some examples of simulated systems are shown in Figure 4.4, exhibiting near-optimal and sub-optimal assemblies.

At an area fraction of $\varphi = 0.04$, the system's equilibrium state is expected to be at fluid-solid coexistence for large enough strength ε_{co} (equivalently at low enough temperature T) [13]. Initiated from a disordered fluid state, it is expected that having $\varepsilon > \varepsilon_{co}$ is not sufficient to guarantee cluster formation due to the long nucleation times exceeding the simulation durations. Instead, only for slightly higher ε is crystallization observed, which lies near the metastable liquid-gas coexistence line [75; 123]. When ε is too large, kinetic trapping is abundant, hindering effective self-assembly within the time scales of the simulation. The yield of the self-assembly is also dictated by our choice of simulation duration t_f , taken to be $t_f = 2 \times 10^7$ VMMC cycles to allow for adequate assembly activity to take place across the range of parameters simulated.

We chose t_f in terms of VMMC cycles and constant maximum step sizes to obtain a consistent self-assembly problem for tuning. The adoption of VMMC over SPMC enables much needed realistic system evolution of cluster morphology via cluster motion, but consistent dynamics for a physically-motivated modelling of the self-assembly were not considered. Aside from the inconsistent timescales of the different moves, the main consequence in our choice of constant Δ_t and Δ_r is the different dynamical regimes accessed by the VMMC algorithm depending on w . Long ranged modes are suppressed for $w > \Delta_r$, limiting large cluster diffusion, whereas internal relaxation of cluster are suppressed for $w < \Delta_r$. As we will see later in the chapter, this does create an artefact in our yield landscape, but nevertheless it provides us with an example system for tuning.

A more physical colloidal system would require a proper treatment of the diffusion coefficients of the translation and rotation moves, correlated with the respective acceptance rates via the definition of Einstein's relations to define a unique timescale. A comprehensive example of a physically motivated treatment of parameter-dependent step sizes for modelling consistent dynamics can be found in a study by Haxton et al. [90], where the maximum step sizes varied with the interaction range to implicitly tune the acceptance rate of the simulations such that they advanced under a consistent unique timescale. The Brownian timescale is commonly chosen to be the unique timescale for self-assembly problems. The Brownian time t_B —the average time it takes for a free particle to diffuse its own diameter—would scale differently with the VMMC time t_V at different parameter state points. Note that these systems are still valid thermodynamically regardless of the evolution dynamics.

4.3 Measures

In this section, we define the quantitative measures used in our study to measure the ability of a patchy particle system to yield large round compact honeycomb clusters. Recall our definition of a *cluster* as an energetically-bound collection of disks interacting with at least one other member via a patch-patch interaction. We choose to use the bond-orientational order susceptibility χ_6 , a measure of the correlations between the ϕ_6 of particle pairs, to quantify the extent of long-range crystalline order, allowing one to assess the size of the

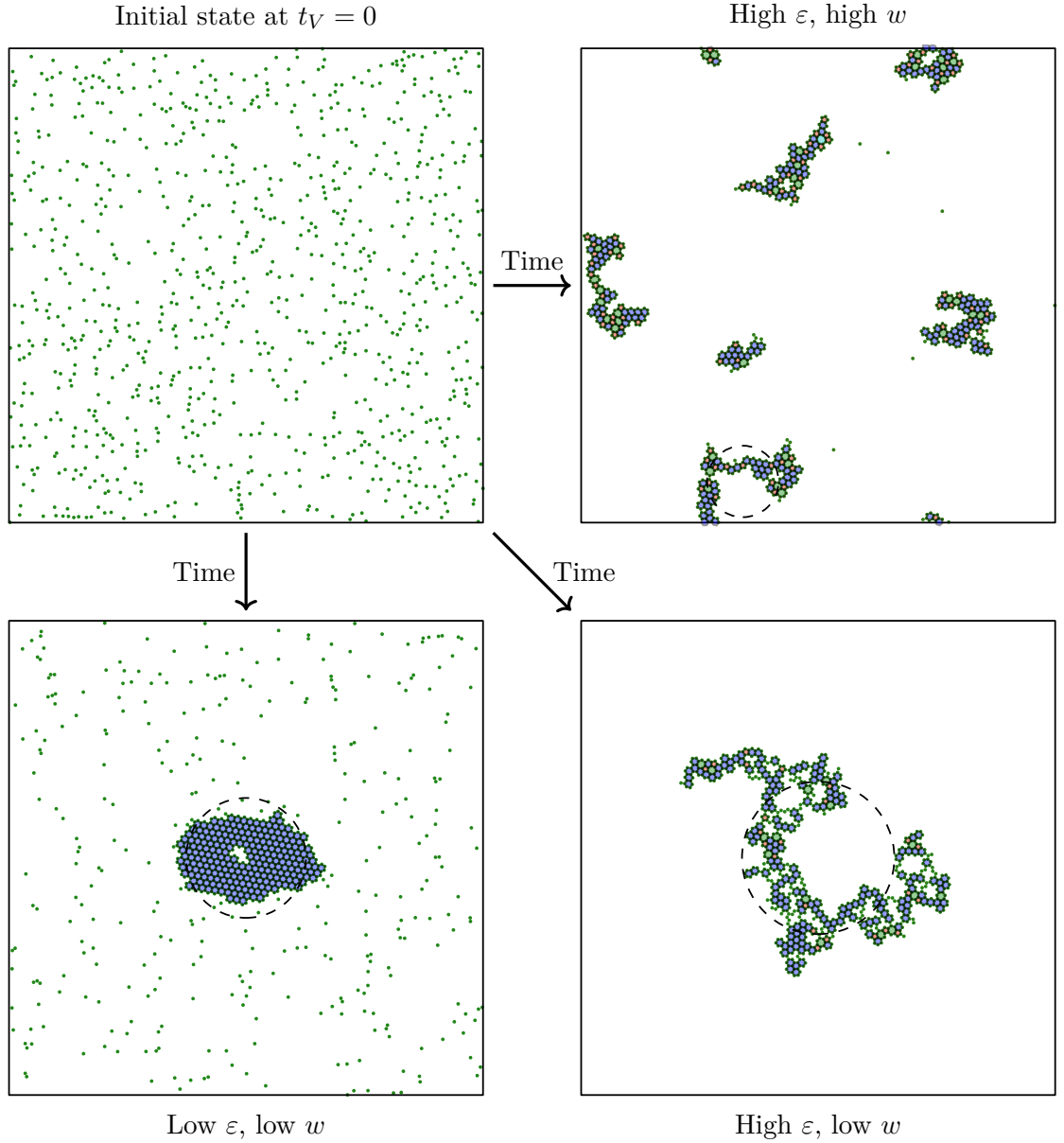


Figure 4.4: VMMC simulations of patchy particle systems initiated from a well-mixed disordered state (top left) and relaxed for $t_V = t_f = 2 \times 10^7$ VMMC cycles. Bottom left shows near-optimal assembly of a single large compact honeycomb lattice. Sparse networks, polygon defects and presence of competing clusters (top and bottom right) all contribute to sub-optimal assembly. Due to the constant rotation maximum step size Δ_r , clusters in systems with $w < \Delta_r$ move relatively freely compared to $w \geq \Delta_r$ systems. Cluster-cluster agglomeration events are therefore rare in the latter case, producing systems with multiple clusters at final times (top right) unlike the former case where single clusters are formed (bottom right).

crystalline domains.

Aside from the associated bulk phase of a cluster structure that forms (e.g. crystal-, gel-, liquid- clusters) which we will refer to as the *quality* of a cluster, we are interested in the *morphology* of a final assembled cluster. If we think of the local particle neighbourhood features (e.g. arrangements and vacancies in lattice/network sites) of the solid as the microstructure of a cluster, then large scale morphological features, including size, shape, symmetry and voids/defects, form the macrostructure or external morphology of a cluster. All macrostructural features apart from cluster size are scale-independent (independent of the cluster size), measured using dimensionless quantities. These quantities often characterise the degree of difference from a target morphology. The target morphology does not necessarily have to be one with the lowest free energy. A compact cluster suggests the efficient arrangement of particles, reducing the space occupied by a cluster network. Inefficient arrangement is interpreted as gaps in a cluster—both locally (such as vacancies in the lattice) and on larger scales (such as space enclosed by a ring-shaped cluster).

Since we are interested in the interaction parameter values that yield the largest clusters, the yield of a system is based upon identifying the cluster with the largest mass (number of particles) in the box, which we call the *principal cluster* C , and performing measurements on C alone. In the following section we define a few measures considered to characterise the system and its principal cluster C . For a consistent definition of round compact target clusters, the set of ideal honeycomb cluster structures H^* for all sizes N_C is established. A compound yield measure Q is then constructed by combining the cluster-specific bond-orientational susceptibility $\chi_{6,C}$ with a circularity-compactness measure N_I/N_C , allowing Q to consider the quality and morphology of the largest cluster found in the system. Several candidate measures were considered which are discussed in more detail below. Observations of our self-assembly systems using these measures can be found in Section 4.4.

4.3.1 Local bond-orientational order and bond-orientational order susceptibility

The local structure around a disk j is defined using the local bond orientational order $\phi_l(j)$ [123]. $\phi_l(j)$ provides an orientation-independent method of quantifying local arrangements of neighbours for each particle j , where l is an integer value of the rotational symmetry in question. We choose $l = 6$ to assess the six-fold symmetry of the local neighbourhood. $\phi_6(j)$ for a particle j with neighbours $j' \in N_b(j)$ is defined as

$$\phi_6(j) = \frac{1}{N_b(j)} \sum_{j'=0}^{N_b(j)} e^{i6\theta_{jj'}}, \quad (4.3)$$

where $\theta_{jj'}$ is the angle between an arbitrary chosen but consistent reference axis and the displacement vector between each neighbour-particle pair j and j' . A local distribution satisfying six-fold symmetry will have $|\phi_6(j)| = 1$, and $0 \leq |\phi_6(j)| < 1$ otherwise. We define $N_b(j)$, the set of neighbours j' of j , as those disks j' whose centres lie within a distance of $a + \delta$ from the centre of j , whether or not a patch-patch interaction exists. By

accounting for disks that lie in close proximity to a cluster C but are not a member of C , one includes the influence of trapped colloidal particles and nearby particles/clusters into the definition of cluster quality.

As mentioned before, long-ranged correlations of similarly oriented particles is synonymous with the formation of large crystal domains. To analyse this larger scale order in C , interparticle correlations in $\phi_6(j)$ within the local structure are considered and integrated spatially [43] to give the bond-orientational susceptibility, defined as

$$\chi_{6,C} = \frac{1}{N_C} \left\langle \left| \sum_j^{N_C} \phi_6(j) \right|^2 \right\rangle, \quad (4.4)$$

where N_C is the number of disks in C . If particles j and j' belong to the same domain then $\phi_6(j) \cdot \phi_6(j') \approx 1$, whereas $\phi_6(j) \cdot \phi_6(j') \approx 0$ between differently oriented members in C .

Although the honeycomb has three-fold symmetry, the three-fold bond-orientational susceptibility $\chi_{3,C}$ would be zero due to the odd symmetry in the lattice cancelling out the contributions. Fluid and honeycomb phases are hence indistinguishable using $\chi_{3,C}$. $\chi_{6,C}$ can also be used to estimate the typical crystalline domain size in C . It is bounded above by N_C ($\chi_{6,C} = N_C$ means a pure crystalline domain), and below by 0 (due to low local ordering and conflicting subdomain orientations). The subscript C in $\chi_{6,C}$ reminds us that this represents how far a local orientation persists throughout the network of the principal cluster, with almost no effects from competing clusters (surface effects still contribute).

Note that under our $N_b(j)$ definition this measure is influenced by any particles lying within suitable distance from C that are not necessarily interacting via patches, e.g. free particles. If all particles in the system are considered then this measure is defined as a system-wide measure, $\chi_{6,sys}$.

4.3.2 Shape factor (compactness-circularity measure)

We define a cluster to be compact if its particles maximise the possible number of bonds while minimising the average length of the shortest paths between all particle pairs in the network. This is equivalent to achieving a minimum average shortest path distance[†] of a network structure in graph theory. The most compact crystals maximise the correct lattice structures formed, while reducing the volume of space occupied. Vacancies in the lattice, appearing as defects within the cluster (Fig 4.5 (D),(F)) or as cavities on the surface of a cluster (Fig 4.5 (E),(F)) contribute to a less compact shape. We identified the symmetric concentric hexagons as the ideal compact cluster shape, compatible with the honeycomb lattice (e.g. Fig 4.5 (A),(C)). Analogous to a disk (the most compact 2D shape) this ideal cluster shape also sufficiently satisfies our target criterion of round clusters. In summary, concerning the cluster morphology, round symmetric clusters are preferred, while clusters with defects are unfavourable.

[†]The distance between two nodes is the number of edges in the shortest path i.e. connected edges, between the nodes [124].

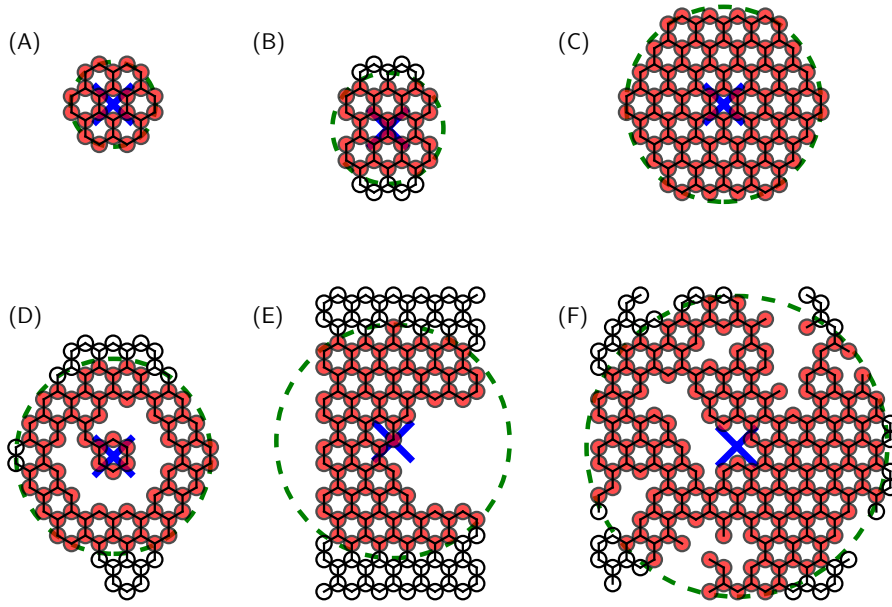


Figure 4.5: Examples of honeycomb cluster networks showing how cluster morphology can differ from the ideal case. The ideal circle S (green dashed circle) derived from constructing the respective mass-equivalent ideal honeycomb $H^*(N_C)$ is shown, centred at the centre of mass (COM) (blue cross) of each cluster. Particles lying within S are filled in red. Clusters A and C are ideal cases $H_{sym}(q = 1)$ and $H_{sym}(q = 3)$; cluster B show mild elongation; a ring-like void is present in cluster D; cluster E demonstrates non-circular shape and cavity in the cluster; cluster F is non-compact with many voids and loose particles not part of any closed loops. The shape factors attempting to quantitatively capture these morphological differences are presented in Table 4.1.

To construct a compactness-circularity measure—we will call this the *shape factor*—we chose the symmetric honeycomb shape to be our target shape and quantified the deviation from this target shape. This measure ought to be independent of the cluster size, but since we are working with particles, the shape factor is more sensitive for smaller clusters. Three candidates were considered to quantify the similarity between C and our ideal target cluster structure: (1) radius of gyration; (2) centre of mass (COM) based overlap measure; (3) coordination number. We first define the target cluster morphology for our self-assembly yield to enable comparison of assembled clusters with the mass-equivalent ideal version.

Target cluster—constructing our ideal honeycomb cluster

The honeycomb cluster satisfying the criteria of being round and compact is the symmetric honeycomb cluster (e.g. Figure 4.5 (A) and (C)). Treating the disks centres as vertices, regular hexagons are considered for the constructed cluster lattices. Starting from a single hexagonal loop of disks, labelled as $H_{sym}(q = 0)$ (green hexagon in Fig. 4.6a), additional hexagons are placed on each of its six sides to form the first ring of hexagons, labelled as $H_{sym}(q = 1)$ (green hexagons in Figs. 4.6c). q tracks the number of concentric rings of

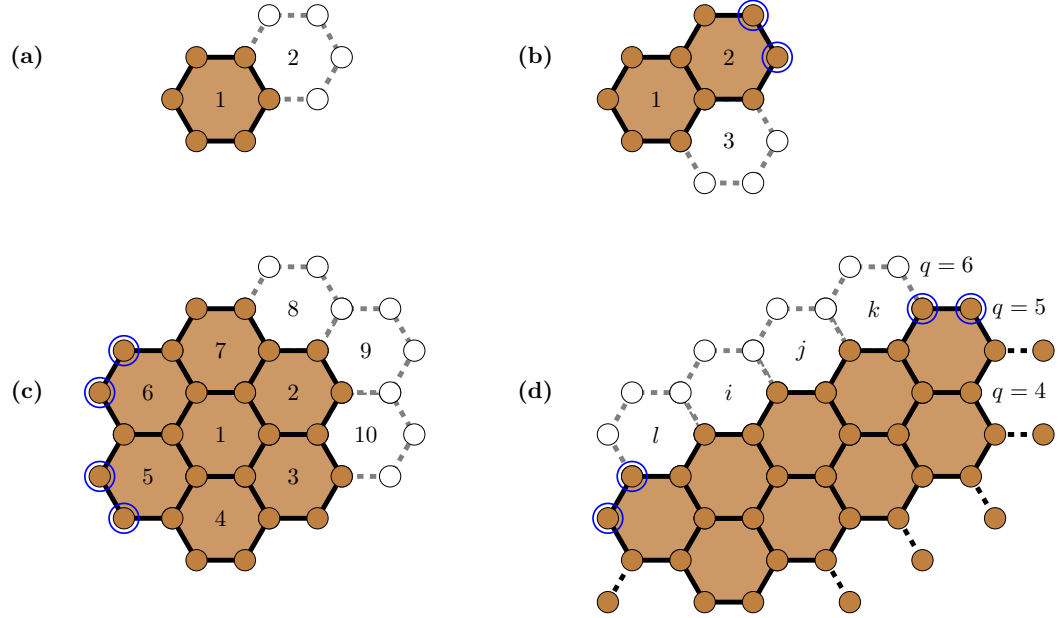


Figure 4.6: Construction of the ideal honeycomb lattices H^* that maximises the number of hexagons (hexagonal loops) created from a given number of particles. The primary rule for creating a new hexagon is to choose one that requires the least additional particles. (a) Starting from a seed hexagon #1 (green filled hexagon) of particles (green filled circles) with bonds (solid edges), a second hexagon is constructed from four bonded new particles (hollow circles) bonding (dashed) with two existing ones. (b) The third hexagon can be constructed using three extra particles. Continuing in this fashion yields a full $q = 1$ shell on which the next hexagon is added. (c) It requires four additional particles to create the hexagon #9 if the hexagon #8 was not created first (which only requires three additional particles itself), hence #8 is prioritised. After #8 is formed, hexagon #9 is created before #10 (both require three additional particles to form)—we use #8 to seed the ‘growth’ of the layer. The rest of the layer is built clockwise following the direction of the growth. (d) Upon the completion of a layer, a face is chosen and the next hexagon is constructed on the next layer from the centre of the face while accounting for symmetry. In this case, the $q = 5$ layer is completed, and the hexagon construction moves onto the next layer ($q = 6$). Multiple sites i, j, k, l are available to construct the next hexagon at. Site i is chosen as it is one of the closest to the centre of the selected face. Working symmetrically from the face centre outwards, subsequent hexagons are added until the face sites are filled, at which point hexagons are added in a clockwise direction from the right end of this face layer until the $q = 6$ layer is complete. Blue circles highlight some example "corners" of H^* where the disks are furthest away from the COM of hexagon #1.

hexagonal loops in the ideal cluster H_{sym} . Subsequent layers are built by adding more hexagons to the surface of the outermost shell. This set of symmetric honeycomb clusters $H_{sym}(q)$ is used as the guideline for our target structure; we try to quantify the similarity of a cluster C to a member from this set. The number of hexagonal loops in $H_{sym}(q)$, which we denote as $|H_{sym}(q)|$, follows the relation

$$|H_{sym}(q)| = 3q(q + 1) + 1, \quad (4.5)$$

and is constructed from $6(q + 1)^2$ particles.

Of course, H_{sym} is only defined for a subset of cluster sizes N_C . To define an ideal target cluster H^* for every N_C , we exploit the fact that using the most efficient approach possible to construct the maximum number of hexagons for a given number of disks yields the most compact honeycomb arrangements. We base our choice of the next hexagon loop to construct by the number of particles required to create it, opting to construct a loop that calls for the least number of particles. Particles closer to the centre of the seed hexagon are added first, with no preference when multiple options are available due to symmetry. New layers are added starting from a single face and built symmetrically from the face centre outwards, before the remainder of the layer is completed in a clockwise manner. Remainder particles are not considered in H^* but contribute to N_C . Details of this ideal hexagon construction are found in Fig. 4.6.

From this set of instructions, one finds that the set of regular concentric hexagon tessellation $\{H_{sym}(q) \mid \forall q\}$ is a subset of $\{H^* \mid \forall \text{ number of hexagons}\}$. In fact, H_{sym} are the unique configurations for their respective sizes, i.e $H^* \equiv H_{sym}$ when $|H^*| = |H_{sym}|$. All other structures are intermediate states between the H_{sym} clusters. We note that $\{H^*\}$ is not necessarily the set of most compact ideal honeycomb clusters for all cluster sizes. Alternative approaches to constructing H^* will affect their centre of mass, which may influence the measurement of its shape factor.

Radius of gyration

The cluster radius of gyration is a geometric measure describing the spatial distribution of mass about the aggregate's centre of mass, capturing its size [125]. For a cluster of N_C equal-mass particles, it is the root mean square of the particle distances \mathbf{r}_i from the cluster's centre of mass

$$R_g^2 = \frac{1}{N_C} \sum_{i=1}^{N_C} (\mathbf{r}_i - \mathbf{r}_{mean})^2 + a_1^2, \quad (4.6)$$

where the centre of mass (COM) is

$$\mathbf{r}_{mean} = \frac{1}{N_C} \sum_{i=1}^{N_C} \mathbf{r}_i, \quad (4.7)$$

and a_1 is the characteristic monomer size. The a_1 component ensures that the dependence of R_g on N_C is non-zero for small clusters: $R_g = a_1$ for a monomer. We take this to be

the particle radius, $a_1 = a/2$ [125].

Particles distributed further away from the COM will increase R_g . Densely packed particles will have a smaller R_g . Consequently, clusters that are more compact have lower R_g . We construct a shape measure s_{rog} by comparing the radius of gyration of a cluster C of size N_C to that of its mass-equivalent ideal target $H^*(N_C)$,

$$s_{rog} = \frac{R_g^*}{R_g}, \quad (4.8)$$

where the label $*$ refers to the ideal cluster. If C is an ideal cluster already, then $s_{rog} = 1$. Clusters that are less compact will have $s_{rog} < 1$, and $s_{rog} > 1$ if they are more compact.

COM-based overlap measure

This measure is inspired by the circularity (or isoperimetric quotient) measure, a quantity gauging the ratio between the area A and square of the perimeter P of a shape, A/P^2 (for a circle this is $1/4\pi$) and subsequently scaled so that the circularity has a maximum of 1^\dagger . For all shapes of area A , the disk has the smallest perimeter. By definition this perimeter is a circle S of radius R_S . Denote the area of the overlapping region between two (filled) shapes as $A_{overlap}$. For a shape and a disk of areas A with their centre of masses (COM) aligned, $A_{overlap} = A$ if and only if the shape is a disk. Otherwise $A_{overlap} < A$ (equivalently, the area of shape lying outside S is greater than 0).

We will define a quantity $s_{overlap}$ to measure the proportion of particles in a cluster lying within R_S from the COM of the cluster. For a cluster of size N_C , $H^*(N_C)$ is used to define a radial bound S with radius R_S . This radial bound is then applied on the cluster C to identify particles that lie within distance R from its COM. We use a circle bound over a hexagon as the ideal shape to target because it is isotropic and easy to implement.

Consider an arbitrary cluster C with N_C patchy particles. To calculate this measure, we first deduce $|H^*(N_C)|$. Considering only the symmetric honeycomb clusters $H_{sym}(q)$ for now, the furthest particle centre \mathbf{r}_{corner} from the COM of the seed hexagon is found at the "corners" of $H_{sym}(q)$ (See Fig. 4.6), thus an imaginary circle $S(q)$ centred at the COM \mathbf{r}_{mean} with radius $R_S(q) = |\mathbf{r}_{corner} - \mathbf{r}_{mean}|$ is the smallest circle able to contain the entirety of $H_{sym}(q)$.

Let $a' = (a + \delta/2)/2$ be the lattice parameter between two neighbouring points. With $b = \sqrt{3}a'/2$, $R_S(q)$ scales with the number of shells by the relation

$$\begin{aligned} R_S(q)^2 &= \frac{a'^2}{2} + (2q+1)^2 b^2 \\ &= \frac{a'^2}{4} + \frac{3a'^2}{4} [4q^2 + 4q + 1] \\ &= a'^2 [3q^2 + 3q + 1] \\ &= a'^2 |H_{sym}(q)|. \end{aligned} \quad (4.9)$$

The terms in the bracket in the penultimate line is equivalent to Eqn. (4.5). Adapting to

[†] Dividing A/P^2 by the ratio for a circle, the circularity becomes $4\pi A/P^2$. For a circle, this will be 1.

clusters of different sizes, we interpolate Eqn. (4.9) linearly for different $|H^*|$ to obtain

$$R_S(N_C) = a' \sqrt{|H^*(N_C)|}, \quad (4.10)$$

which we use to define the radius of the corresponding ideal circle $S(N_C)$.

Drawing $S(N_C)$ on the original cluster C , centred at its COM, the overlap measure is calculated by taking the ratio

$$s_{overlap} = \frac{N_I}{N_C}, \quad (4.11)$$

where N_I is the number of disk centres that lie within the space enclosed by $S(N_C)$.

Coordination number

The final candidate measure considered is based on the coordination number, the number of nearest neighbours of a particle. The coordination number describes the local behaviour of a cluster solid, and is useful for identifying crystalline behaviour in packing problems of isotropic colloids where the expected average number of interactions is known. Only considering patch-patch interacting neighbours for each particle, the coordination number n_3 of a cluster of size N_C is the average over the coordination number $n_{3,i}$ of all particles i in the cluster, i.e. $n_3 = (1/N_C) \sum n_{3,i}$. Although $\langle n_3 \rangle$ cannot differentiate the network quality in our system (particles participating in fully connected polygonal or honeycomb lattices have $\langle n_3 \rangle = 3$), it is correlated with the size of a maximally-connected domain. Members of the cluster that are not maximally-connected are found in defects (voids in the structure and grain-boundaries) and on the cluster surface. By definition, all ideal honeycomb clusters $H^*(N_C)$ maximise n_3 for the respective cluster size N_C , as the number of bonds in the cluster is maximised when forming the most hexagon loops possible.

To turn n_3 of a cluster of size N_C into a shape factor we again compare it to the coordination number n_3^* of the corresponding $H^*(N_C)$ to give

$$s_{coord} = \frac{n_3^*}{n_3}. \quad (4.12)$$

Performance

Examples of different cluster shapes are shown in Figure 4.5, with their corresponding shape factor measurements given in Table 4.1. Clusters (A) and (C) are H_{sym} clusters with $q = 1$ and 3 respectively, and all candidate measures give a shape factor of 1. Cluster (B), a slightly elongated cluster, is compact but not as circular—most candidate measures give values less than 1. Clusters (D),(E) and (F) demonstrate how voids in the lattice, cavities on the surface, elongated shapes, and fractal-like solids all contribute to low compactness and circularity. All measures can detect these features, but we found $s_{overlap}$ to be the most sensitive to deviations from the H_{sym} cases. Although a circle is used as the guide in $s_{overlap}$ to approximate the deviation of clusters from the hexagonal H^* shape, in practice, this measure is good at distinguishing compact and round shapes from poor ones with voids.

Table 4.1: Table summarising shape factor candidates $s_{overlap}$, s_{rog} and s_{coord} for the example clusters found in Figure 4.5 and 4.7, along with the measure they are derived from. Clusters C and D have the same size ($N_C = 96$), but the latter is less compact and circular due to the void in its structure. All candidate measures are able to differentiate this, with $s_{overlap}$ exhibiting the largest change. Overall, $s_{overlap}$ is the most sensitive to the deviations from the ideal honeycombs H^* , while s_{coord} is the least sensitive.

| | N_C | $ H^*(N_C) $ | $R_S(N_C)$ | N_I | $s_{overlap}$ | R_g | s_{rog} | n_3 | s_{coord} |
|------|-------|--------------|------------|-------|---------------|-------|-----------|-------|-------------|
| A | 24 | 7 | 2.78 | 24 | 1.00 | 0.95 | 1.00 | 2.50 | 1.00 |
| B | 38 | 12 | 3.64 | 28 | 0.74 | 1.11 | 0.92 | 2.58 | 0.99 |
| C | 96 | 37 | 6.39 | 96 | 1.00 | 1.02 | 1.00 | 2.75 | 1.00 |
| D | 96 | 36 | 6.39 | 75 | 0.78 | 1.20 | 0.85 | 2.54 | 0.92 |
| E | 134 | 47 | 7.64 | 83 | 0.62 | 1.32 | 0.81 | 2.69 | 0.97 |
| F | 210 | 57 | 9.85 | 175 | 0.83 | 1.19 | 0.91 | 2.55 | 0.90 |
| XYXX | 66 | 20 | 5.14 | 34 | 0.52 | 1.19 | 0.86 | 2.58 | 0.96 |
| XYXY | 66 | 20 | 5.14 | 42 | 0.64 | 1.09 | 0.93 | 2.58 | 0.96 |

s_{coord} is the least sensitive to the different morphological features, since no positional distribution information of the particles is contained in the measurement. Figure 4.7 demonstrates one major disadvantage in using s_{coord} over s_{rog} and $s_{overlap}$. Taking two copies of two different domains (X—the 9-ball, Y—a single loop), clusters XYXX and XYXY are formed by the arrangement of these domains that they are named after as shown. Both clusters are of the same size and have the same number of hexagonal loops in each. Different arrangements of congruent subdomains lead to different shapes, affecting the overall form of the cluster. Intuitively we identified cluster XYXX as being less compact than XYXY. s_{coord} fails to distinguish between these different arrangements.

On the other hand, s_{rog} and $s_{overlap}$ show a difference in the two arrangements due to the differing COM positions. The inclusion of the cluster COM in the definitions offers a reference point to study the spatial distribution of the cluster network. Vacancies within the structure and concave features on the surfaces occupy space but do not contribute to the honeycomb network, increasing the spatial range covered by the cluster. Disks lying further away from the COM push the cluster boundaries outside of S , and increase the radius of gyration of the cluster. We found that $s_{overlap}$ places more emphasis on having the particle distribution closer to the COM, and consequently the roundness and compactness of the cluster. For this reason, we select $s_{overlap} = N_I/N_C$ over s_{rog} as our shape factor for the yield measure Q .

4.3.3 Compound yield measure

So far we have looked at measures that capture the size of honeycomb domains in clusters and the shape of clusters, satisfying the different objectives we desire in our final duration assembly product. To obtain a compound yield measure Q of the principal cluster C in the system, we combined the bond-orientational susceptibility $\chi_{6,C}$, shape factor N_I/N_C

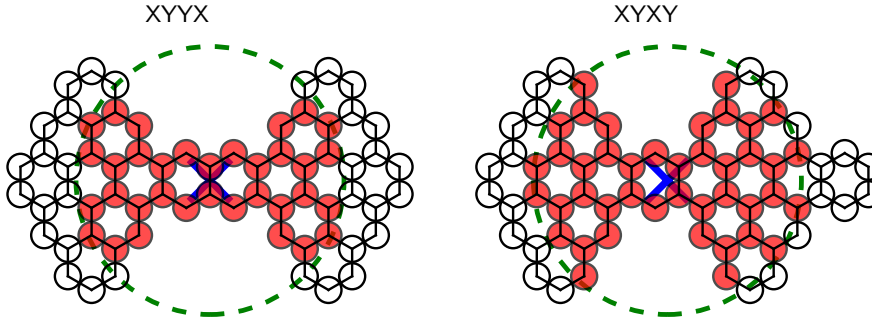


Figure 4.7: Cluster networks XXXX and XYXY contain the same number of disks and hexagon loops, but differ in the order of subdomains X (9 hexagons arranged in 9-ball configuration) and Y (single hexagon). The different particle arrangements lead to different COM (blue crosses), resulting in different N_I , the number of disks (red) lying within the ideal circle guide S (green dashed circle), and in the radius of gyration R_g . These two cluster are indistinguishable when only considering coordination number ($s_{coord} = 0.9551$ for both), but s_{rog} and $s_{overlap}$ recognise that XYXY is more compact and circular than XXXX (See Table 4.1 for numerical data).

and cluster size N_C in a multiplicative form

$$Q = \frac{\chi_{6,C}}{N_C} \left(\frac{N_I}{N_C} \right)^\alpha N_C, \quad (4.13)$$

where α is chosen as a weight factor between the different aspects. We express the yield measure with explicit N_C appearing in both the numerator and denominator to emphasis the three aspects that form the target criteria for the structure ($\chi_{6,C}/N_C$ captures the structural network/phase of the cluster; N_I/N_C the morphology; N_C the size). Although a large honeycomb lattice is desired, this yield measure ensures that the cluster morphology is not compromised.

Targeting interaction parameter values that exhibit quality self-assembly outcomes defined on multiple design criteria (quality, size and shape in this case) constitutes a multi-objective optimisation problem, where multiple tests are used to measure the performance (the fitness) of the candidate values (the solutions). By combining the measures together we leave ourselves with a single yield quantity to work with—a single-objective optimisation problem with only a single test of fitness. Combining multiple objectives can help reduce the dimensionality of the problem, and reduce the size of the solution set by removing extreme solutions (for example a solution that satisfy one set of design criteria but is considered a major outlier in another criteria would require only one test instead of several to determine that is a poor one) [126]. The multiplicative form of our definition of Q achieves this; the non-linearity of Q means clusters with balanced performance in quality and size ($\chi_{6,C}$) and shape (N_I/N_C) have a chance to rank higher than large clusters with poor shape for example. Q is of the same dimension as N , and scales with the cluster size N_C .

4.3.4 Summary

We have described several measures characterising different aspects of our self-assembly system. In order to target good assembly yield, we adapted existing definitions (bond-orientational order) and new measures (circularity/compactness measure) in order to fully express the target structure sought. $\chi_{6,C}$ formed the basis of the yield measure to identify the formation of sizeable honeycomb clusters. This was then modified using a shape factor N_I/N_C to filter those products that have poor morphology, resulting in a single compound yield measure that we will use to evaluate the effectiveness of the patchy disk system to assemble the target structure.

N_I/N_C is sensitive to small deviations from the ideal target shape, placing emphasis on particles distributed near the centre of mass of the cluster. It is very specific to the system studied as it measures the deviation from the target ideal H^* , but demonstrates how cluster morphology can be explicitly incorporated into our design criteria. This shape factor measure is more suited for computational studies, with the task of finding N_I for a physical system being difficult.

4.4 Self-assembly observations

We report the simulation outcomes for our patchy hard disk model simulated under the conditions described in Section 4.2.

4.4.1 Assembly yield at final time

A heatmap plot of the expected yield $\langle Q \rangle_{t_f}$ on the ε - w parameter space is shown in Figure 4.8, with $\alpha = 1.4$ to emphasise the cluster shape more (see Section 4.4.2). The yield for each state point (ε, w) on the plot is an arithmetic average of the final time yield of 10-16 independent simulations, represented by the notation $\langle \cdot \rangle_{t_f}$. An uneven distribution of simulations were performed during the mapping of Q as large areas of low yielding state points were subsequently identified after an initial set of simulations. This is not dissimilar to standard practices when mapping parameter space, where intermittent user input is common to determine whether a subset of state points has been identified already (in this case at around the top right and bottom left corners of the parameter space where large flat plateaus are found), in which case no further data collection for the respective points is necessary. We restricted the range of w to 20° or less as the thermodynamic equilibrium state prefers disordered polygonal networks at high w [13]. Example clusters depict (A) compact honeycomb with few polygonal defects, (B) compact honeycomb, (C) polycrystalline glass, and (D) polygonal glass. A region of significant yield activity extends from high strength ε and low flexibility w along the lower half of the plot, curving upwards towards low ε and high w to form a pointed peninsula. The yield on this peninsula can be visualised as a hill with a global peak at $\mathbf{x}^* = (8.6, 4.0)$ near (B), being heavily skewed towards the low ε, w side of the peninsula and gradually fading in the positive ε, w direction.

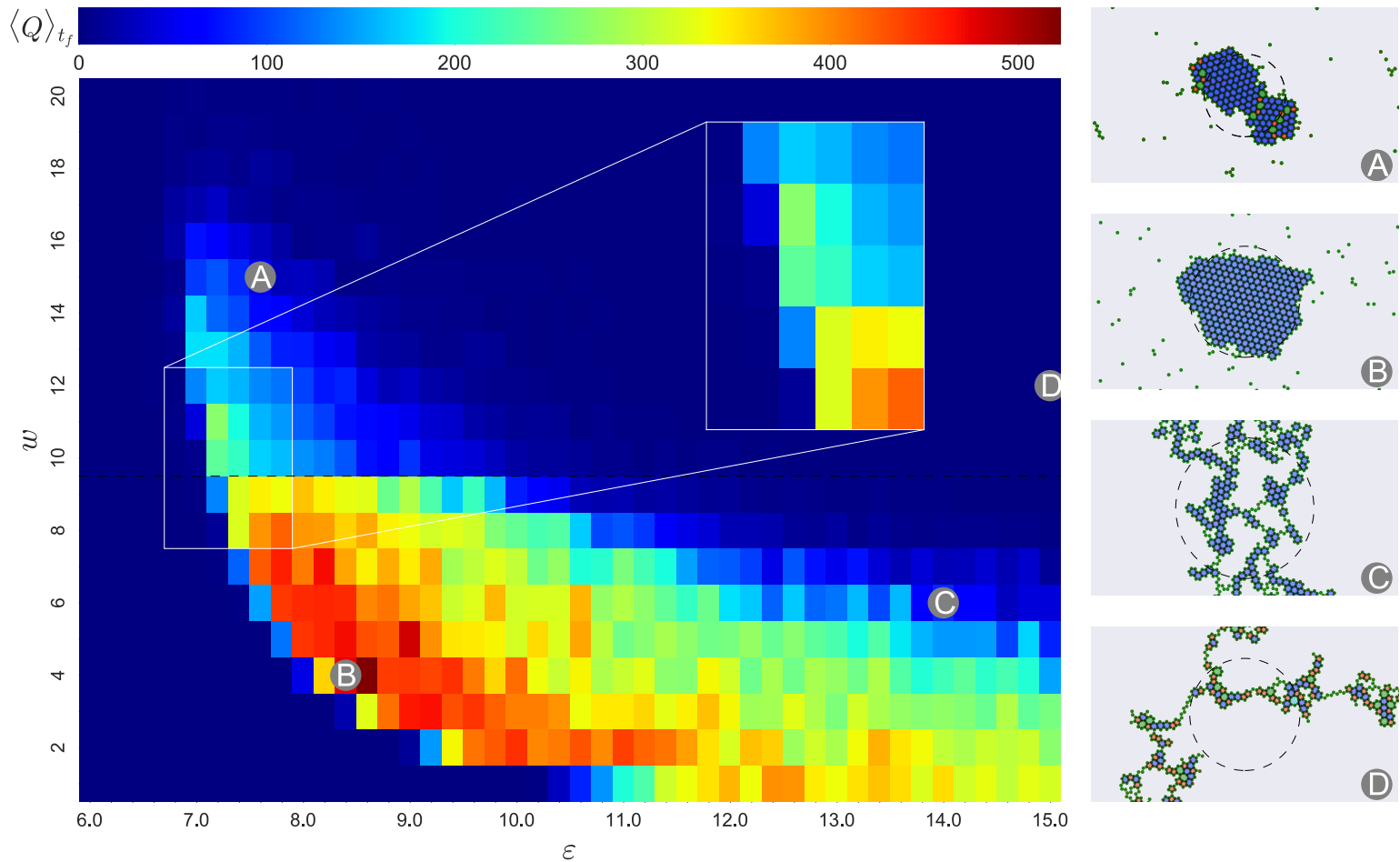


Figure 4.8: Heatmap of the compound yield $\langle Q \rangle_{t_f}$ with $\alpha = 1.4$ averaged over 10-16 simulations each, at final time $t_f = 2 \times 10^7$ VMMC cycles on the ϵ - w parameter space. The global maximum of 522.7 (1.d.p) is located at $\mathbf{x}^* = (8.6, 4)$ in the region of near-optimal assembly. The dashed line at $w = 10^\circ$ divides the parameter space into the two different dynamical regimes modelled. Example principal clusters have coordinates: (A) (7.6, 15), (B) (8.4, 4), (C) (14, 6), (D) (15, 12). Inset shows the secondary local peak with significant prominence at (7.2, 11).

The secondary peak at (7.2, 11) highlighted in the inset is an artefact of the discretisation of the parameter space \mathcal{P} that create local maxima. The artefact occurs when a ridge of the yield surface—a chain of local maxima forming a continuous curve—passes at an angle to the coordinate axes such that points $b_1, b_2, \dots \in \mathcal{P}$ lying sufficiently close enough to the ridge are not all neighbours. The points in between some of these b_1, b_2, \dots have relatively low $\langle Q \rangle_{t_f}$, creating local maxima. This secondary peak is not the only local maxima on Q created in this way (for example, this also occurs at the high ε , low w region), but it has a high prominence and will have a significant effect on the hill-climbing behaviour in Chapter 6.

At low ε (high temperature T), a fluid state exists where no assembly activity occurs. Moving to higher ε (lower T), nucleation occurs readily within simulation durations, indicated by the dynamical phase boundary on the lower side of the peninsula. Optimal assembly, where relatively circular single domain honeycomb clusters form such as example cluster (B), was found at within two $\delta\varepsilon$ greater than this dynamical boundary for each w , near the metastable liquid-gas coexistence curve [75; 123]. A larger ε is required to stabilise the more specific (low w) interactions [67; 75]. Disks become increasingly able at forming non-hexagonal closed loops as their patches widen, so polygon loop defects in the final yield structure also become more common, e.g. cluster (A). Lowering the interaction strength ε provides better annealing behaviour to counter the increased tendency of polygon loop formation, highlighting the importance of bond reversibility for good assembly [67; 68]. The ridge of significant yield at high ε low w is expected to extend further along the ε axis.

Increasing ε further for each fixed w leads to a rise in kinetically trapped features in the system, as mentioned in Fig. 4.2. We identified two categories of kinetic defects contributing to poor assembly: polygonal and structural. *Polygonal defects*, as the name suggests, are non-hexagonal loops embedded in the network as disks efficiently bond to each other (yielding coordination number $n_3 = 3$) but does not maximise their rotational entropy, predominantly affecting $\chi_{6,C}$. We categorise other defects such as voids in the lattice, grain boundaries, disk chains etc. that prevent an efficient bonding of disks as *structural defects*. Structural defects play a bigger role in determining the macrostructure of the cluster.

The number of polygonal defects in a system increases as w is increased, with fewer polygonal loops annealing out along increasing ε . On the other hand, a rise in structural defects occur when ε is increased as faster bonding rates do not allow adequate time for disks to arrange themselves into loops. In the large ε limit, the resultant clusters resemble the products seen in diffusion-limited aggregation, where particles undergoing Brownian motion irreversibly agglomerate to a cluster, forming fractal-like trees [127]. The interplay between these two types of defects yields compact clusters at low ε (example clusters (A) and (C)) and low-density glassy aggregates at high ε (Cluster (C) and (D)) with more polygonal defects as w is increased.

The region of significant yield owes its peninsula shape to $\langle \chi_{6,C} \rangle_{t_f}$ (see Fig. 4.9a). Low $\chi_{6,C}$ is indicative of the lack of significant honeycomb domain growth. The largest honeycomb domains ($\langle \chi_{6,C} \rangle_{t_f}$ peak) are yielded at (11.8, 2.0). For each w , a non-monotonic

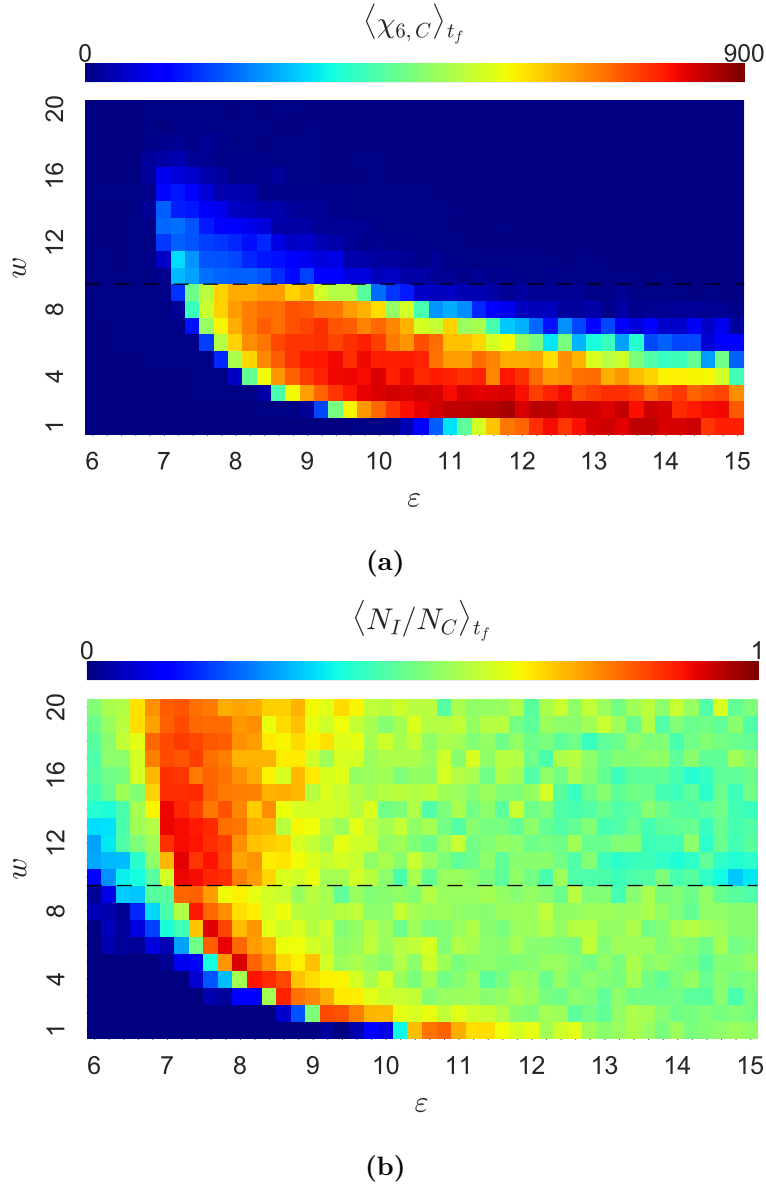


Figure 4.9: Heatmaps of bond-orientational susceptibility $\langle \chi_{6,C} \rangle_{t_f}$ and shape factor $\langle N_I/N_C \rangle_{t_f}$. $\langle \chi_{6,C} \rangle_{t_f}$ reveals that the largest honeycomb domains are found in systems with high ε and low w . High shape factor coincides with near-optimal target assembly, where predominant particle agglomeration rather than cluster-cluster agglomeration leads to round compact growth.

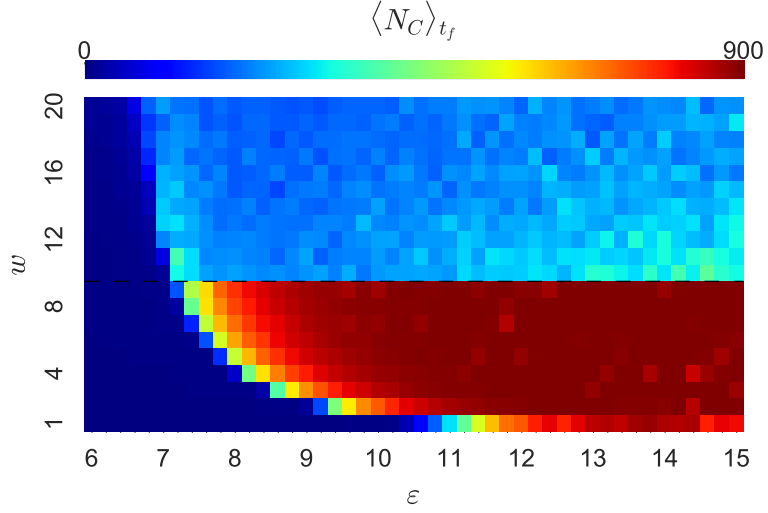


Figure 4.10: Heatmaps of $\langle N_C \rangle_{t_f}$. Simulations show two distinct regions of cluster growth below and above the dashed line at $w = 10^\circ$. For $w < 10^\circ$, $\langle Q \rangle_{t_f}$ shows a monotonic increase on ε and w , as stronger interactions enable faster growth of clusters. For $w \geq 10^\circ$, clusters in the system fail to merge within simulation time t_f , resulting in smaller principal clusters. Consequently, the two regions have distinctly different magnitudes in $\chi_{6,C}$ and N_I/N_C along w axis.

profile of $\langle \chi_{6,C} \rangle_{t_f}$ against ε is observed as expected [81] (similar in form to Fig. 4.16a). Starting from the peak interaction strength along a fixed- w profile, weaker interactions lead to smaller honeycombs (see Fig. 4.10), but stronger interactions raises the levels of structural defects found in principal cluster.

Figure 4.9b shows the shape factor $\langle N_I/N_C \rangle_{t_f}$ of the final time principal clusters. Although a larger honeycomb domain is yielded at high ε low w regions, the cluster also deviates away from our target structure as it becomes increasingly elongated (the cluster circularity decreases). The roundest and most compact clusters (high N_I/N_C) are located in a narrow band of parameter space along the region of optimal assembly. For $w > 10^\circ$, this band widens along the ε axis as w is increased due to the increased flexibility of patches enabling easier loop formation and thus higher network distortion, helping the network to stay compact. The discontinuity in the heatmaps is due to the two different dynamical regimes modelled by the VMMC algorithm. We discuss this in more detail in Section 4.4.3.

4.4.2 Effects of tuning α

Table 4.2 on page 52 lists the top 5 yielding state points for $\alpha = 1.4$, shown alongside $\langle Q \rangle_{t_f}$ for $\alpha = 1.0$ and $\alpha = 2.3$. The heatmaps in Figure 4.11 show the corresponding yield landscapes. Numerically, the principal clusters found at $\varepsilon = 8.6$, $w = 4.0^\circ$ consistently yielded the highest global $\langle Q \rangle_{t_f}$ for a large range of α . After visually inspecting the principal clusters, we confirmed that the products assembled in the neighbourhood around $\mathbf{x}^* = (8.6, 4.0)$ did indeed reflect the best yielding target structure. However, we found that with no tuning ($\alpha = 1.0$), high ε , low w systems yielded elongated honeycomb clusters

Table 4.2: The top 5 best yielding state points ordered for $\langle Q(\alpha = 1.4) \rangle_{t_f}$ (shown in bold). Although the clusters found at $\mathbf{x}^* = (8.6, 4)$ are not the largest in size N_C , they are well-balanced in shape and quality. Compare this with the state points $(11.8, 2.0)$ and $(8.0, 5.0)$ with the highest $\langle \chi_{6,C} \rangle_{t_f}$ and $\langle N_I/N_C \rangle_{t_f}$ respectively (italicised). For all α tested from 1.0 to 2.5, \mathbf{x}^* consistently gives the best yielding assembly behaviour. All values are to 1 or 2 decimal places.

| ε | w/o | $\langle \chi_{6,C} \rangle_{t_f}$ | $\langle N_C \rangle_{t_f}$ | $\langle N_I/N_C \rangle_{t_f}$ | $\langle Q(1.0) \rangle_{t_f}$ | $\langle Q(1.4) \rangle_{t_f}$ | $\langle Q(2.3) \rangle_{t_f}$ |
|---------------|-------|------------------------------------|-----------------------------|---------------------------------|--------------------------------|--------------------------------|--------------------------------|
| 8.6 | 4.0 | 672.3 | 697.6 | 0.84 | 561.0 | 522.8 | 447.3 |
| 9.0 | 5.0 | 791.1 | 835.8 | 0.70 | 556.5 | 484.9 | 357.8 |
| 8.4 | 4.0 | 579.2 | 601.3 | 0.86 | 501.2 | 473.9 | 419.7 |
| 8.4 | 5.0 | 690.3 | 727.3 | 0.76 | 521.4 | 468.4 | 371.4 |
| 8.2 | 7.0 | 697.9 | 765.0 | 0.75 | 522.5 | 468.1 | 369.6 |
| 8.0 | 5.0 | 499.9 | 527.4 | <i>0.92</i> | 460.2 | 445.7 | 415.6 |
| 11.8 | 2.0 | <i>867.2</i> | 892.3 | 0.51 | 448.6 | 350.8 | 208.0 |

that were ranked too high, with the contribution of $\chi_{6,C}$ outweighing N_I/N_C . Increasing α raised the weighting of the macrostructure contribution to Q . As a consequence, the peak at the near-optimal assembly region around \mathbf{x}^* became more prominent. However, we felt that too much emphasis was placed on the macrostructure over the honeycomb domain quality by the time a value of $\alpha = 2.0$ is reached. We used an arbitrarily chosen value of $\alpha = 1.4$ between these bounds as our final choice in order to favour cluster morphology enough to achieve more reasonable prominence in the neighbourhood of \mathbf{x}^* .

4.4.3 Differences between the two dynamical regimes of the VMMC algorithm

Recall from Section 4.2 that we chose a constant maximum step size $\Delta_r = 10^\circ$ for VMMC rotational moves for all w , resulting in two categories of dynamical regimes modelled by the VMMC algorithm: dynamics exhibiting adequate collective motion when $w \geq \Delta_r$, and dynamics with long range correlated motion suppressed when $w < \Delta_r$. The effect of having two dynamical regimes across the parameter space led to distinct differences in the self-assembly outcomes, especially in aspects concerning the size and morphology of the clusters. This can be seen in many of the heatmaps as the discontinuity at $w = \Delta_r = 10^\circ$, highlighted by the red dashed line.

The different dynamical regimes modelled by the VMMC do not have a significant bearing on the thermodynamical outcome of the assembly, as suggested by the absence of this discontinuity in the internal energy $\langle |E| \rangle_{t_f}$ in Figure 4.12. It does, however, have huge consequences on the size of the principal cluster C , as seen in Fig. 4.10. Consequently, $\chi_{6,C}$, which measures the size of the honeycomb domains, and Q also show a discontinuity. The discontinuity seen in $\langle N_I/N_C \rangle_{t_f}$ in Figure 4.9b is a result of late time cluster-cluster merging (see Section 4.4.5).

For the $w > 10^\circ$ region, a slight increase in $\langle N_C \rangle_{t_f}$ is present in the neighbouring state points of $(15, 10)$. As ε increases and w decreases, the clusters formed become less compact, thus covering a larger area. The larger sphere of influence means an increased likelihood

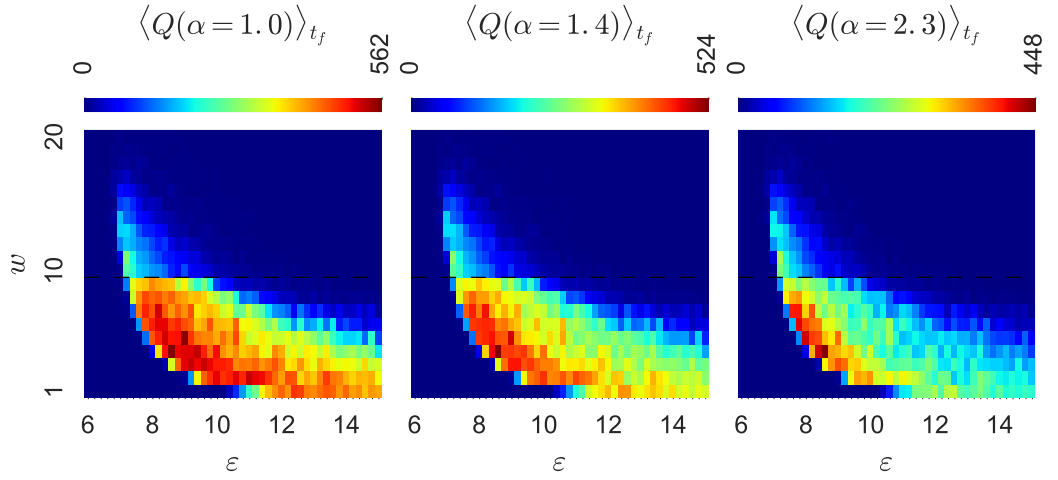


Figure 4.11: $\langle Q \rangle_{t_f}$ heatmaps for $\alpha = 1.0, 1.4$ and 2.3 . Note the color gradient scale is normalised to the maximum of the respective landscape. Increasing α favours the assembly behaviour of honeycomb clusters that are also rounder and more compact. This is found at lower ε systems, enhancing the prominence of the global $\langle Q \rangle_{t_f}$ peak at \mathbf{x}^* .

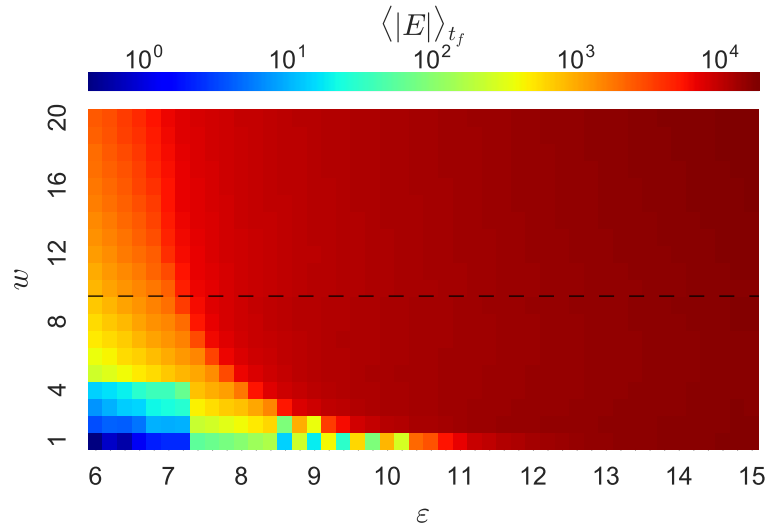


Figure 4.12: Heatmap of the magnitude of the total energy $\langle |E| \rangle_{t_f}$ of the system in logarithmic scale. The absence of a ridge at $w = 10^\circ$ indicates that the different dynamical regimes has little bearing on the thermodynamical outcome of the self-assembly.

of a cluster interacting with a neighbouring cluster in the system, reflected by the slightly larger $\langle N_C \rangle_{t_f}$ values. Upon successful agglomeration, the resulting cluster is less compact and more elongated (see Section 4.4.5), hence a drop in shape factor $\langle N_I/N_C \rangle_{t_f}$ is also seen in the corresponding region in Figure 4.9b.

4.4.4 Closed loop formation

One primary indicator of high quality assembly is the creation of hexagonal closed loops: tessellating hexagons form the honeycomb. The abundance of other polygon loops indicate poor aggregate structures. The evolution of the system-wide polygon counts N_z over t_V for polygon loops of size z is shown for various different interaction strengths ε for $w = 8^\circ$ (Fig 4.13) and $w = 13^\circ$ (Fig 4.14). In both cases, increasing ε leads to an increase in polygonal defects. The increase in structural defects for high ε results in a decrease in total number of loops at final time.

Non-hexagonal loop formation is disfavoured when $w = 8^\circ$, but form more easily when $w = 13^\circ$. We suspect the significant N_5 growth for $w = 13^\circ$ systems at early times is due to the rate for a chain of 5 particles manoeuvring into a pentagon loop being faster than the rate at which an additional particle agglomerates onto an awaiting 5-particle chain to form a hexagon loop. This depletes the opportunities for hexagon loops to grow early on. Pentagon formation and the annealing out of these defects therefore play an increasingly significant role in the assembly pathways of honeycomb networks as w is increased, in agreement with Ref. [128].

4.4.5 Cluster growth dynamics—effects on the cluster morphology

As monomers collide and bind via attractive patches, clusters form and grow via the agglomeration of particles and clusters. Morphological studies of cluster formation via the agglomeration of monomers have been studied in Ref. [125] from solving Langevin equations of motion in a continuum regime. Modelled as short-ranged isotropic interacting monomers, it was found that at early times monomer-cluster agglomeration was the dominant growth mechanism of clusters, whereas cluster-cluster agglomeration occurred at late times. The resulting clusters were found to be compact yet elongated, with these two characteristics being respectively attributed to the isotropy and relatively short range of the monomer-monomer potential.

Similar behaviour was observed in our system. Multiple nucleation events leading to cluster growth occurred for high ε systems. At low ε only a few clusters grew at any one time, if any. Steady growth is achieved from the agglomeration of monomers and small clusters throughout the system's evolution, producing round clusters. For high ε , multiple clusters grew during the simulation to significant size. Then, at late times, cluster-cluster agglomeration becomes the main mechanism for large cluster growth, which, in the absence of free particles and good annealing, produces elongated clusters and aggregates with abundant structural defects. As touched upon in Section 4.4.3, the two dynamical regimes realised by the algorithm thus resulted in different levels of cluster-cluster agglomeration in the simulations, affecting the shape and size of the principal cluster (See Figs. 4.9b

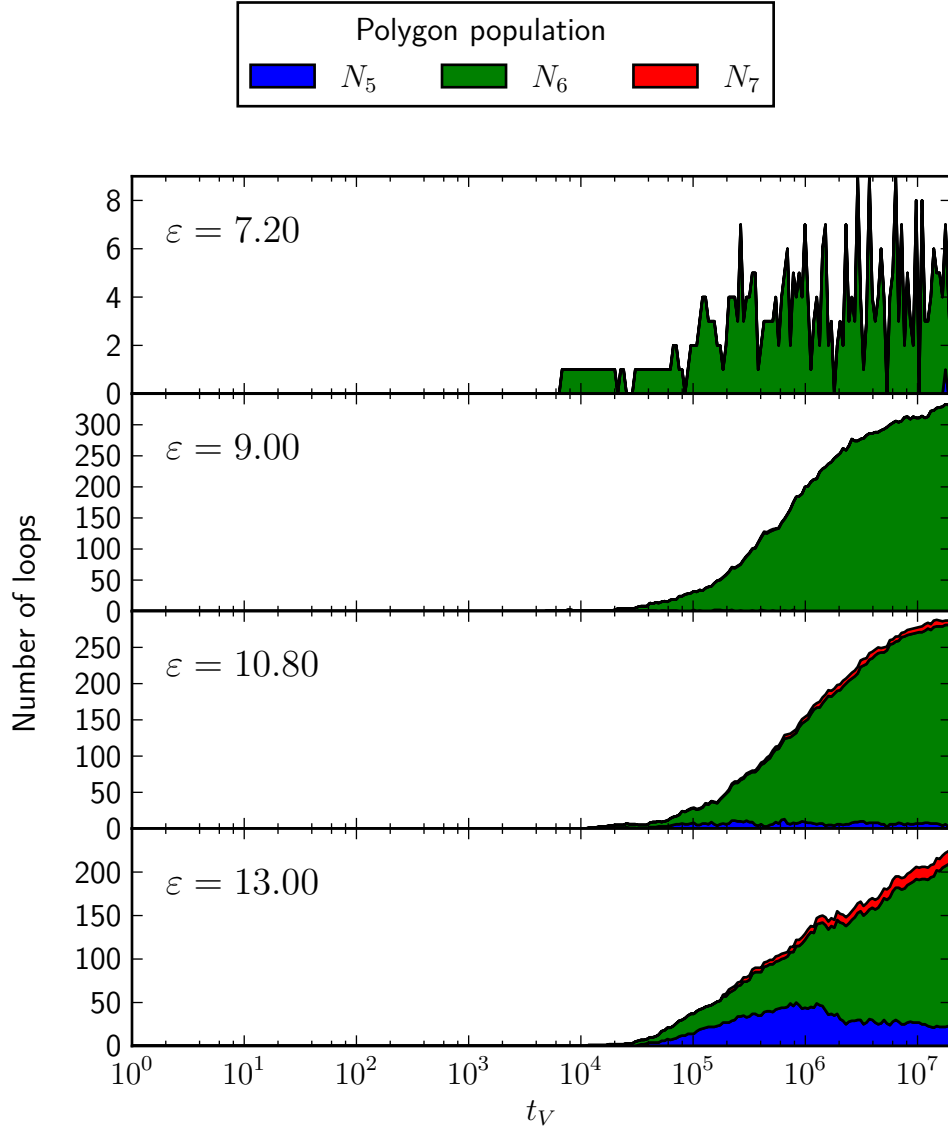


Figure 4.13: Stacked time trajectories of system-wide polygon population for $w = 8^\circ$ systems at various strengths ε . The polygon loop counts N_s , defined on a logarithmic VMMC time t_V , are shown for loop size s from 5 to 7 (Other polygon counts are negligible). All strengths started growing at similar times, suggesting that nucleation is not a rare event for this range of ε . At $\varepsilon = 7.2$, the interactions were not sufficiently strong, resulting in a low number of polygon closed loops forming. At $\varepsilon = 9$, N_6 dominated as honeycomb domains formed with little kinetic trapping. At $\varepsilon = 10.8$, increased levels of kinetic trapping led to the stabilisation of pentagons N_5 and heptagons N_7 . By $\varepsilon = 13$, pentagon loops did not anneal out fast enough, hindering hexagon formation. The decrease in N_5 after $t_V = 10^6$ indicates the annealing of pentagon loops into hexagon loops. The rise in structural defects from increasing ε led to a lower total number of loops.

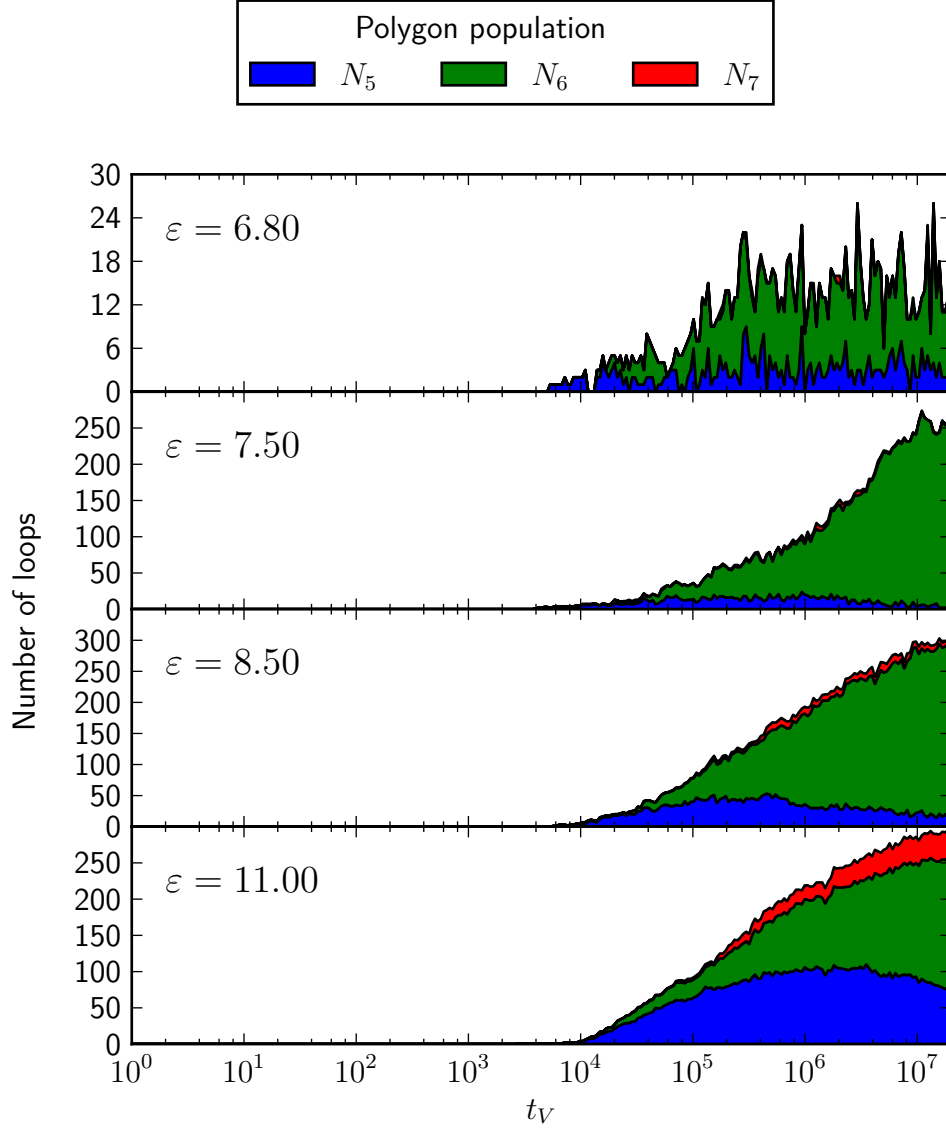


Figure 4.14: Stacked time trajectories of system wide polygon population for $w = 13^\circ$ systems at various ε . The polygon loop counts N_s , defined on a logarithmic VMMC time t_V , are shown for loop size s from 5 to 7 (other polygon counts are negligible). N_5 was more significant compared to Figure 4.13 as wider patches enabled non-hexagons loops to form and persist more easily. Interactions were too weak at $\varepsilon = 6.80$ to form many loops. Overall polygon population increased with ε , with hexagon loop formation dominant across most ε . Pentagons played an increasingly significant role in the earlier stages of assembly as ε was increased. At $\varepsilon = 7.5$ and 8.5 , enough bond forming and bond breaking activities enabled healthy N_6 growth while N_5 was kept low. By $\varepsilon = 11$, bonds were so strong that the assembly was dominated by pentagon loops before annealing out into hexagon loops.

and 4.10 on pages 50 and 51). For $w < 10^\circ$, where cluster motion is significant, cluster sizes reach the system size limit $N_C = N$ as competing clusters in the system merge to form a single aggregate. However, these principal clusters become increasingly distorted and elongated in shape. For $w \geq 10^\circ$, where long range diffusion of clusters is suppressed, inter-cluster motion is poor and becomes increasingly so as N_C increases, limiting cluster-cluster agglomeration within the simulation times observed. Numerous clusters remain in the system at final times and hence N_C remains relatively small, while clusters have a large N_I/N_C over a wider range of ε values.

We clarify this with an example. Figure 4.15a shows a example N_C trajectory for a low w system near optimal assembly, with N_I/N_C tending towards 1 as C grows. Note that in general N_C does not necessarily track the same cluster throughout the assembly, although manual inspection indicates that most of this example trajectory tracks the same C . A significant merging event between two clusters of similar size occurred at some time between $t_b = 8126497$ VMMC cycles (Fig. 4.15b) and $t_a = 8629358$ VMMC cycles (Fig. 4.15d), the latter marked with a dot-dashed line. This merging event is reflected by a significant decrease in N_I/N_C (Fig. 4.15c as a result of two round clusters merging into an elongated one).

The presence of late time cluster merging events thus increases the level of large scale structural defects in C . Not only does this explain the discontinuity in the N_I/N_C heatmap, it also explains the sudden narrowing of the high N_I/N_C band at the $w = \Delta_r$ boundary; for systems at sufficiently high ε where numerous clusters grow, reducing w to below Δ_r led to an immediate increase in the number of cluster merging events as cluster motion was enabled. The qualitative agreement of the observations here with Ref. [125] suggest that the presence of cluster motion is a sufficient requirement to realising late-time cluster-cluster agglomeration activity for short-ranged interacting hard particles, with the interaction strength determining the level of cluster-cluster agglomeration activity in the system.

4.4.6 Summary

We presented results from the VMMC simulations of the patchy hard disk system, characterising its self-assembly behaviour. The range of self-assembly activity observed in \mathcal{P} reflect the diversity in self-assembly activity observed in Ref. [13]. The interaction parameters ε and w influence the level of polygonal and structural defects to yield poor assembly, near-optimal assembly or kinetically trapped structures. The yield parameter Q successfully distinguished between the various outcomes, combining aspects of local and correlated ordering with the cluster macrostructure to quantify the similarity between the assembled principal structures and our target structure. Importantly, the state point satisfying our target design criteria produced the global maximum Q value.

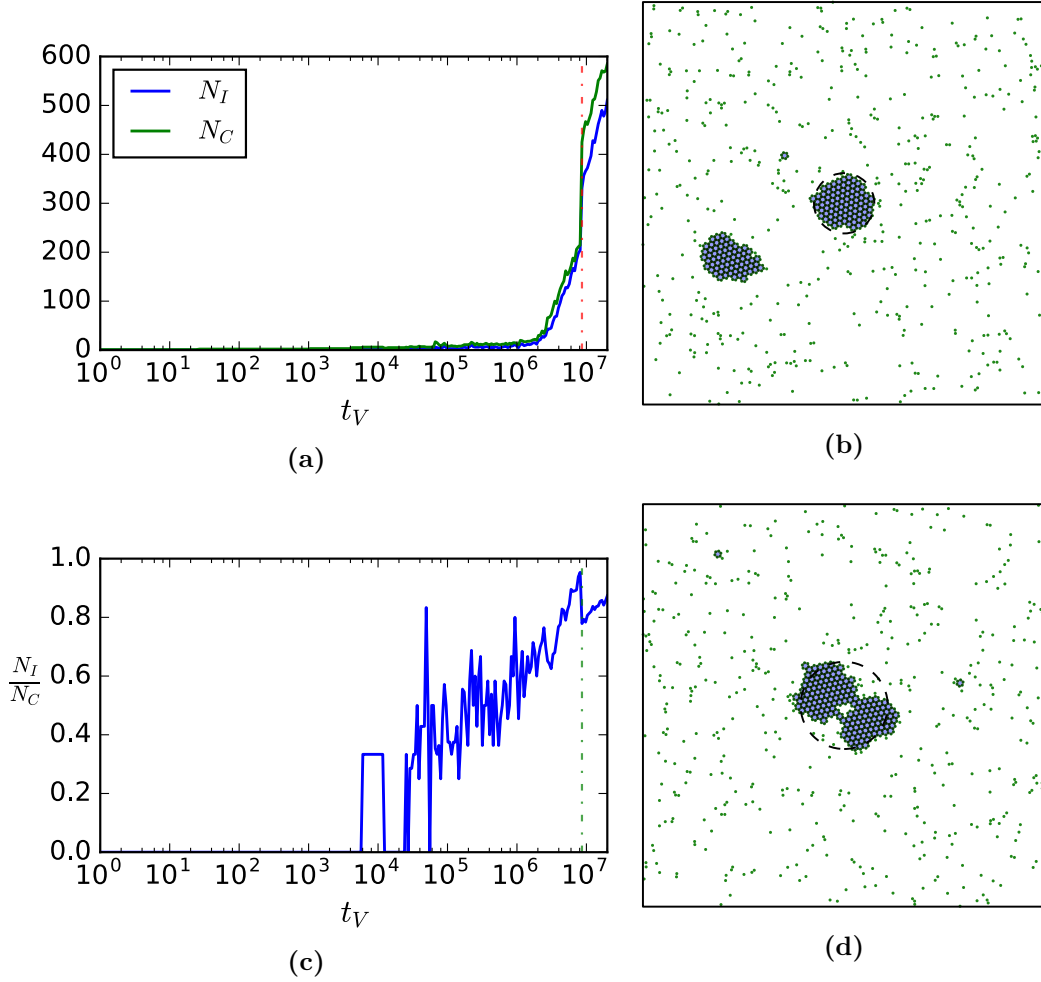


Figure 4.15: Example of effect of growth and merging of clusters on the shape factor. (a) shows N_I and N_C trajectories for an example system with $\varepsilon = 8.4$ and $w = 4^\circ$ over simulation time t_V . (c) shows the shape factor N_I/N_C trajectories. On the right are snapshots of the system (b) before a cluster merging event at $t_b = 8126497$ VMMC cycles and (d) after at $t_a = 8629358$ VMMC cycles. The principal cluster C is shown with its ideal circle $S(N_C)$. Between times t_b and t_a (highlighted by the dot-dashed line), two medium sized round clusters merge to form one larger cluster that is elongated and contains structural defects. This is reflected in the graphs as a large jump in N_C and N_I , but a drop in shape factor N_I/N_C .

4.5 Yield landscape properties

Our yield landscape exhibits properties that can be expected in self-assembly yield landscapes of other systems. We discuss some of these properties important for characterising our optimisation problem.

4.5.1 Time dependency

The yield landscape $\langle Q \rangle_{t_V}$ is dependent on the assembly duration t_V , affecting the scale of the yield values at each state point \mathbf{x} differently as it increases by varying rates according to the assembly pathways explored (Fig. 4.16a). The different assembly behaviours also result in a form-changing yield landscape whose yield peak location \mathbf{x}^* is non-stationary.

This time dependency carries over to the full 2D parameter space. Figure 4.16b is a plot of two averaged yield trajectories for two points $(10, 4)$ and $(8.6, 4)$, along with the global maximum $\langle Q \rangle$ value at each recorded duration t_V . The patchy disk simulations at $(10, 4)$ proceeded with clusters forming quickly very early on, but non-optimal assembly led to an eventual slow-down in Q growth after $t_V = 4 \times 10^6$ VMMC cycles. Simulations at $(8.6, 4)$ exhibited an initially slow but steady growth in Q , becoming the global peak by $t_V = 1.2 \times 10^7$ VMMC cycles. The $Q(t_V)$ peak (black dashed trajectory) passes through these two state points, from high strength where cluster size growth is rapid with large $\chi_{6,C}$, to lower strength where the cluster is consistently round and compact.

4.5.2 Noise in the landscape: fluctuations in the measurements

The yield landscape describes the *expected* assembly outcome of the system at each state points, with the consistency of the self-assembly outcomes reflected in the variance σ^2 of the expected yield. The yield definition determines the magnitude and change in variance across the landscape. Figure 4.16c, showing the expected Q for $w = 5^\circ$, exhibits a large standard deviation σ at high Q throughout due to the high variability in cluster shape. Compare this with the same profile but for $\chi_{6,C}$ in Figure 4.16d. Here σ is small in the near optimal assembly region, reflecting the consistency in size of the gradual honeycomb growth. The consistency of assembly outcomes (reproducibility) and its reflection in the yield measure is an important aspect of the self-assembly particle design problem to consider.

In the next half of this thesis, our study on the practicality of using local search heuristics to optimise the assembly yield of the patchy model will involve taking advantage of these fluctuations in the measurements. The handling of the fluctuations in the assembly outcomes is also relevant to the intensity of the search heuristic, as each step of the search requires the evaluation of the self-assembly performance of the patchy model system, hence increasing the accuracy of $\langle Q \rangle$ demands a higher computational intensity. We attempt to keep this computational intensity low, which is useful when one already needs to the computational investment for each complex particle simulation is already large.

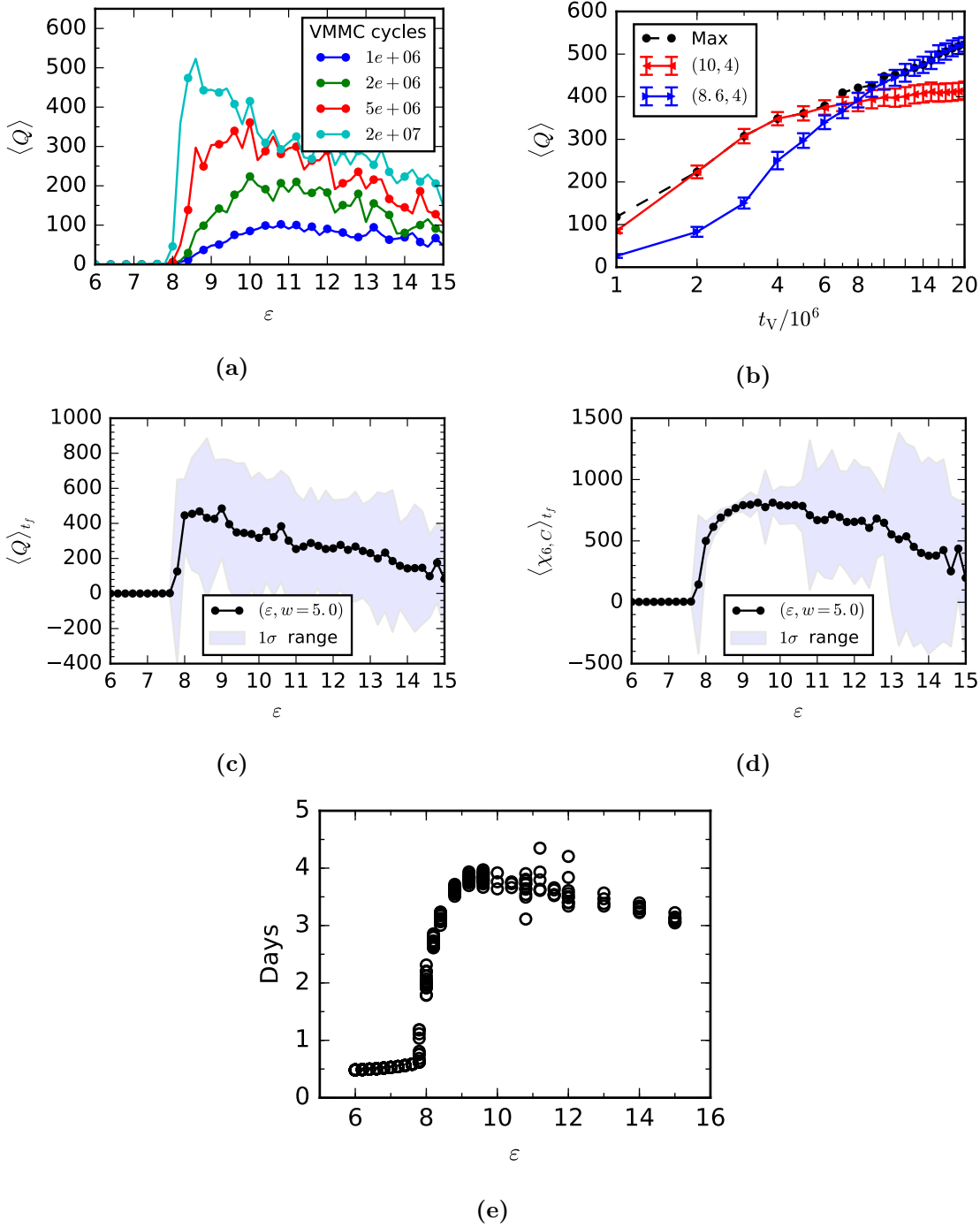


Figure 4.16: (a) Profile of $\langle Q \rangle$ against ε for $w = 4^\circ$, at various simulation times t_V , depicting the characteristic form of a yield curve. Too weak a bond (low ε) leads to insufficient cluster growth and assembly activity, whereas too strong a bond (high ε) leads to abundant kinetic trapping inhibiting quality structural assembly. Different assembly pathways exhibited by varying ε result in $\langle Q \rangle$ evolving over simulation time at different rates. Some symbols are omitted for clarity. (b) The location $\mathbf{x}^* = (\varepsilon^*, w^*)$ of the global yield maximum (black dashed) is dependent on the simulation time t_V , passing through $(10, 4)$ at early time, where rapid honeycomb growth is found, before settling at $(8.6, 4)$ for $t_V \geq 1.2 \times 10^7$ (c) Final time Q profile for $w = 5^\circ$ with $1\text{-}\sigma$ range band (light blue). (d) Final time $\chi_{6,C}$ profile for $w = 5^\circ$ with $1\text{-}\sigma$ range band (light blue). (e) The execution time t_{EXEC} of VMMC simulations plotted along ε for $w = 5^\circ$.

4.5.3 Computational demand of VMMC simulations

For each simulation of the patchy particle system ran according to the details in Sections 4.2, the execution time t_{EXEC} was dependent on the state point simulated, varying between 0.5 and 5 days to simulate 2×10^7 VMMC cycles (Fig. 4.16e). Each run is a single CPU simulation, hence the total computational demand t_{TOTCPU} is equivalent to the execution time t_{EXEC} (See Section 5.2.2).

The large differences in t_{EXEC} reflect the computational processes of the VMMC algorithm. VMMC attempts to move collections of particles at each step. The computational demand of each step scales with the number of particles pairs considered during the recruitment stage of the VMMC move, thus the average t_{EXEC} per VMMC cycle increases during the simulation as cluster domains grow.

Another situation where a large time difference in t_{EXEC} could arise for self-assembly simulations is when they are simulating according to physical timescales, such as the Brownian time t_B . In this case, the number of VMMC cycles would vary with the state point to keep a constant Brownian time duration.

4.6 Conclusion

We have presented a self-assembly system of patchy particles with interaction design parameters interaction strength ε and patch flexibility w and executed a set of VMMC simulations of the system for a finite range of ε and w . We took the opportunity to deviate from conventional yield definitions found in self-assembly studies based solely on microstructural qualities (and common derivatives such as correlations of local properties): we demonstrated how macrostructural features can be considered by incorporating a shape factor into the yield measure Q and controlling its weighting via a tuning constant α . The gathered data formed a yield landscape $\langle Q \rangle$, with the best yielding parameters at $\mathbf{x}^* = (8.6, 4.0)$ corresponding to the global peak of $\langle Q \rangle_{t_f}$. Consequently, we were able to target honeycomb clusters that are large, round (circular) and compact.

This yield landscape forms the benchmark for our optimisation studies. The simulation data will also form an integral part of the latter half of the thesis where, under the interest of time and statistical significance, a yield landscape model will be constructed based on the data, on which our investigation of hill-climbing and optimisation will be carried out. The yield landscape is noisy and time-dependent. These are general properties that we believe are found in most dynamical self-assembly yield landscapes. We will keep these properties in mind for the next part of the study, which focuses on treating parameter tuning as an optimisation problem.

Chapter 5

Review of local search heuristics and hill-climbing techniques

We now move onto the tuning of interaction parameters for maximising target assembly yield. As mentioned previously we are interested in looking at alternative approaches for exploring parameter space with the primary goal of reducing the computational cost and execution time for finding the best yielding parameter values. The yield landscape $\langle Q \rangle_{t_f}$ obtained in the previous chapter represents the targeted design model we want to tune, and the tuning problem equates to locating high yielding state points near/at the global maximum of $\langle Q \rangle_{t_f}$. Good assembly, associated with a localised range of interaction parameters, is dependent on the assembly duration. One defining feature of the yield landscape that we shall exploit during optimisation is the intrinsic noise of the assembly outcome.

In this chapter we will review the fundamentals and introduce the language of optimisation and hill-climbing, with emphasis on applications to noisy landscapes. We look at hill-climbing, a local search algorithm (heuristic), as an approach to finding interaction parameters leading to optimal assembly of target structures. We choose to focus on hill-climbing since it is easy to implement and adaptable to any colloidal design problem. Then, we summarise existing metaheuristics (problem-independent algorithmic tools) that handle noise in landscapes to control the behaviour of an algorithm, motivating our metaheuristic study of the trial period limit m of a stochastic hill-climbing algorithm in the next chapter.

5.1 Optimisation problems

Let \mathcal{S} be an \mathcal{M} -dimensional set of points and f a function defined on this space, such that for all points $x \in \mathcal{S}$, $f(x)$ returns a value. A simple question to ask is where the maxima of f lie, i.e. the location x of each peak, and more often than not, the global maximum of f at x^* . An evaluation $f(x)$ at all points x will give us the answer—we simply choose x that gives us the highest value—but this approach is highly inefficient. If \mathcal{S} contains an enormous set of points and/or the act of evaluating a point x to obtain $f(x)$ is time consuming and costly, this restricts the practicalities of scanning the full search space. It

is then natural to ask how one can find the global maximum more efficiently, and whether it is necessary to evaluate all available points in \mathcal{S} when only the peaks are of interest. The latter is especially relevant when it is only practically feasible to sample a subset of \mathcal{S} .

This example illustrates the basic setup of an optimisation problem, an approach to solving the optimisation problem along with the consideration of practicality when selecting the method. An optimisation problem is defined by the *search space* of possible solutions \mathcal{S} and the *objective/fitness function* f which represents the degree to which the solution satisfies the criteria numerically. $f(x)$ can be a mathematical evaluation, or a measurement in an experiment or a simulation. The goal of optimisation is to find the optimal or near-optimal solution in \mathcal{S} that maximises/minimises f .

\mathcal{S} can be a discrete set of solutions or a continuous space of points. It is common to identify a subset of solutions in \mathbb{R}^M defined within a finite range satisfying a set of constraints, and for \mathcal{S} to be a grid of points representing a discretised version of this subset to reduce the size of the search space, at the cost of introducing error in the discrete representation. Moreover, search spaces tend to have a notion of distance between solutions which define the relationship between two solutions, with similar solutions having shorter distances. This is typically the Euclidean metric (distance measure) when considering a continuous space of parameters. A search space formed from a discretised subset of this parameter space naturally inherits the same metric. From the notion of distance a *neighbourhood* can be defined, which determines how the solutions are similar to one another. The neighbourhood definition \mathcal{N} , search space \mathcal{S} and objective function f combine together to form a *fitness landscape* $(f, \mathcal{S}, \mathcal{N})$. For convenience we will use f to refer to both the fitness landscape and the fitness function. Without generalisation, we look to maximise f .

The *global optimal solution* (maximum in this case) to the optimisation problem is the overall fittest solution, i.e. the solution $x^* \in \mathcal{S}$ where $f(x^*) \geq f(x)$ for all $x \in \mathcal{S}$. When a notion of neighbourhood is defined, i.e. when concerning fitness landscapes, locally maximum solutions may also exist, where a solution $x \in \mathcal{S}$ is a *local maximum* if $f(x) \geq f(x')$ for all $x' \in \mathcal{N}(x)$. Often the global maximum is the desired solution to problem. However, in the interest of time or flexible operating conditions, finding a local maxima may suffice. When a single maximum is present, the fitness landscape is unimodal. When multiple maxima are present the fitness function is multimodal.

Optimisation methods are applied to search for these maxima. One may use a range of algorithms to explore \mathcal{S} and find x^* and other local maxima. In the example at the start of this section, the enumeration and systematic evaluation of all available points (solutions) in \mathcal{S} describes a *brute-force* method to solving such optimisation problems. Exact methods such as the brute-force search guarantee that the global optimal solution is found and identified. Unfortunately they are often impractical as they don't scale well with the size of the search space or are unfeasible for very complex problems (not polynomial (NP) problems [129]), hence they can only realistically handle problems of small sizes. Approximate methods, known as *heuristics*, are developed to find solutions within a reasonable amount of time, making them practical for most real-world problems.

On the other hand, they do not guarantee that an optimal (local or global) solution will be found.

5.1.1 Heuristics and metaheuristics

Heuristics are approximate algorithms for solving optimisation problems, invented to find the ‘best’ solution to a problem after taking into account the objective of the optimisation and factors limiting the execution of the algorithm. They are approximate because an exhaustive search is not performed and hence the final answer is not guaranteed to be the global optimal, unlike exact methods. Heuristics are often algorithms that, starting with an initial solution, iteratively suggests and assesses new solutions, replacing the current solution if a new solution is deemed better as it progressively converge towards a better solution. When the new solutions are generated from the local neighbourhood of the current solution, the algorithm is a *local search* algorithm. Local search heuristic can be single-solution based algorithms (hill-climbing, gradient descent, simulated annealing [130], tabu search [131]) or population-based algorithms (genetic algorithms) [129]. The above iterative local search description alone yields a hill-climbing if a value-based comparison of the solutions is considered, or gradient ascent/descent if the gradient between solutions is used as the indicator. Ref. [132] provides a classification of optimisation techniques including population based algorithms (such as evolutionary algorithms), local search and global search algorithms.

Since a local search heuristic generates moves locally, it will find local maxima. Heuristics that only accept fitter solutions cannot escape local maxima, so reaching global optimum is dependent on the initial solution. Protocols to assist in the escaping of local maxima generally approach from four categories: considering different initial solutions; accepting non-improving neighbours; changing/varying the neighbourhood; changing/varying the objective function/data [129]. Popular alternatives to hill-climbing consider at least one of these approaches. For example, simulated annealing, developed to emulate the slow cooling of a substance so that it reaches equilibrium crystalline structures, accepts neighbours with lower fitness (non-improving) with a temperature-controlled probability scaling with the fitness difference [130]. Simulated tempering, a cousin of simulated annealing, is used in the exploration of free energy landscapes by treating the temperature as a dynamic variable to ensure the system is kept in equilibrium [133]. Tabu search [131], evaluating the full neighbourhood, accepts non-improving moves if no improving moves exist. It memorises recently visited solutions to avoid revisiting them and can be extended to memorise its history to expand the different decisions made during its search. The noisy method adds artificial noise to the objective function, so that measurements of the fitness fluctuates. This noise is then gradually reduced at each iteration.

The decision for using a particular algorithm for solving an optimisation problem involves considering the features of the landscape (for example ruggedness—the variability of fitness values and their distribution in search space; global landscape structure such as symmetry and modality; dimensionality; noise) and the constraints in execution (execution time, computational resources, etc) in order to tailor the algorithm to suit the needs of

the user [134]. Identifying the complexity of the landscape greatly informs us of the complexity of the search algorithm to deploy, or the rate at which to expect acceptable solutions to be found [135]. Unimodal functions ensure that the global optimum is found when an optimum is located, so local search techniques often suffice. Multimodal functions with rugged surfaces require more thought into ensuring that the search algorithm finds the basin of attraction of the global optima, for example utilising multiple search agents (Particle Swarm technique) to communicate and work together to increase the chances of hitting this peak earlier in the search. An iterative algorithm can be terminated at any time since its solutions are always valid (but not necessarily the best). The termination condition of an algorithm is thus tailorable, and often forms limits for the search parameter choices.

The concept of metaheuristic is varied and still disputed. Heuristics are often borne from tackling an optimisation problem, whose algorithm details are characterised into a distinct routine defining the nature of the heuristic. Many algorithmic aspects of a heuristic can be made variable, or modified for more adaptable purposes in order to raise its performance. Moreover, a heuristic considers many parameters addressing the demands specific to an optimisation problem. The treatment of all these aspects when considered on its own, regardless of the specific problem, forms a set of guidelines and strategies. We will use the term *metaheuristic* as the study of these aspects of a heuristic beyond a specific problem. This includes any general conceptual approach and strategies that can be applied to groups or families of heuristics. Metaheuristics are used as a framework for studying and analysing existing heuristics, as well as for informing and motivating choices when tailoring a heuristic to tackle a new optimisation problem. It may be easier to try to distinguish these two definitions based on why they are of interest. Heuristic research is concerned with finding ways of optimising a problem; metaheuristic research is interested in how to improve current heuristics and inventing new ones. Ref. [136] provides a good history of metaheuristics and the conceptual nuances associated with the unclear division between higher-level metaheuristic framework and lower-level (meta-)heuristic algorithm.

One pair of concepts addressed in metaheuristic studies is the idea of expanding the area of space searched to unexplored areas (*diversification*) and concentrating the exploration in localised regions with high-quality solutions (*intensification*). New (meta)heuristics often spawn from popular heuristics by modifying or adding to existing intensification and diversification metaheuristic protocols. As mentioned earlier, local search heuristics naturally converge to the local maximum in the basin of attraction that the initial solution lives in, which may not be the global maximum. Several adapted metaheuristic techniques manipulate the downhill movement of agents to introduce diversification, enabling climbers to escape from the basin of attraction and enhance the search potential of an algorithm, by using for example artificial noise (simulated annealing), memory (tabu search), and multiple agents (particle swarm) [129].

It is difficult to consider all these modifications and their effects on local search algorithms theoretically. Instead, empirical studies of various strategies are required in order to understand the resulting behaviour. Local search algorithms are widely applicable and typically have low empirical complexity, so they are popular as optimisation tools for tack-

ling many problems. They are used for many intricate and complicated problems precisely for their ease in setup, with few constraints and requirements beside defining a neighbourhood of the search space and an initial solution [137]. The choice of the initial solution may dictate the quality of the final outcome, therefore local search algorithms may be run several times at different initial solutions to increase the likelihood of reaching the global maximum.

5.2 Redefining the parameter tuning problem

The main goal of this thesis is to explore parameter tuning strategies to enhance the efficient formation of the largest, round compact honeycomb colloidal clusters. Interpreted as an optimisation problem, this corresponds to finding the parameter values that maximise the yield of target self-assembly. In the previous chapter we constructed a yield measure Q to capture this targeted assembly objective, with higher Q corresponding to a structure that is more similar to our desired target. Analogous to our simple example in Section 5.1, the interaction strength ε and bond flexibility w form a two dimensional parameter space and Q is the fitness function for the optimisation problem. We take the search space \mathcal{P} to be the discretised subset of this parameter space with spacing $\delta\varepsilon = 0.2$ and $\delta w = 1^\circ$ to obtain a grid, at a range of $6 \leq \varepsilon \leq 15$ and $1^\circ \leq w \leq 20^\circ$, corresponding to the sampled state points in Section 4.4. Other discretised values were not considered due to the restriction of the simulated data. However, we expect the qualitative trends we observe to be consistent regardless of the discretisation. Q , \mathcal{P} and the Euclidean metric forms our yield landscape, characterising the patchy particle system’s predicted behaviour for different interaction parameters at time t_V . Tuning interaction parameters becomes a search of the global optimum $\mathbf{x}^* = (\varepsilon^*, w^*)$ on this Q yield landscape. In practice, the objective of the search could be relaxed to locate a sufficiently high yielding state point in the global peak region around \mathbf{x}^* .

5.2.1 Constraints in simulation-based parameter tuning

To evaluate the yield Q of a state point $\mathbf{x} = (\varepsilon, w)$, the corresponding particle system is simulated using the VMMC algorithm and measured at the end of the simulation. Each independent simulation ran at \mathbf{x} will return a different Q value due to the stochastic nature of the system. We can see this *evaluation* of \mathbf{x} , $Q(\mathbf{x})$, as being *noisy*. The presence of noise in Q complicates the comparison between solutions. Indeed, since the yield function are relatively simple in form without noise, the presence of noise only increases the difficulty of the search [138]. The conventional method for handling noise involves taking the mean of several measurements/samples to obtain an evaluation with reduced noise, at the cost of increased computational demand.

Existing knowledge of the system can help speed up the tuning process. First and foremost is to determine the parameter ranges to consider for the search space \mathcal{P} where good assembly is highly probable. Restricting the size of the initial \mathcal{P} limits the problem size. Temperatures too high or too low set natural thermodynamical bounds where self-

assembly is non-existent/non-ideal at equilibrium. Apart from using knowledge of phase diagrams and behaviour from previous studies of related systems, analytical and numerical methods can be used to identify and eliminate infeasible regions of good assembly. Our decision to limit the w range to 20° or less (See Section 4.4.1) is an example of this. In literature, zero-temperature genetic algorithm search [51; 99] was applied to determine the preferred zero temperature equilibrium structures and infer finite temperature structures. Techniques such as connectivity landscape analysis [139; 140] looked to identify the subset of parameter space which satisfied a set of geometric constraints, further reducing the domain size where good assembly takes place. What remains after identifying \mathcal{P} is to simulate particle systems in this parameter region to find the best yielding region.

Our concern in methodology is in reducing the total computational cost and/or execution time required to satisfy the search objective. Each MC simulation requires long times and large computational demand in order to reach the defined cutoff time of the self-assembly problem, making it computationally expensive to call an evaluation. The brute-force search is useful for mapping out and understanding phase behaviour defined on the interaction parameters [75; 90; 91], with the optimal or near-optimal solution being a by-product of the conducted scan. When the main interest in studying the system is to locate the optimal or near-optimal parameters that enhance the targeted self-assembly yield, sampling the full search space \mathcal{P} is inefficient and only practical for small \mathcal{P} . When \mathcal{P} is large, it is preferable to only sample a subset of the solutions in order to reach the answer. We apply hill-climbing to our tuning problem to examine its ability and speed to locate high yielding interaction parameters.

5.2.2 Measuring search performance

With the choice of search algorithm to apply dependent on the problem studied, no unique standardised measure of search efficiency exists to characterise the performance of a search algorithm. Our concerns when applying a search heuristic often involve the length of execution time and amount of resource spent performing the search. The execution time is the amount of time that the user waits for a simulation or search process to run. For simulations and algorithms, the resource is taken to be the computational effort, usually measured in computational time.

Various timescales are considered to evaluate the performance of a computational task (simulation or search process). The *CPU time* t_{CPU} is the amount of time used by a central processing unit (CPU) to perform a task. Simulations may exploit parallel-processing, in which case two or more CPUs are utilised for a single computational task. Here the total computational demand required to execute the task is taken: the total CPU time t_{TOTCPU} is the sum of the CPU time spent by each CPU. We extend this definition to include the total CPU time required to run a set of tasks.

The *execution time* t_{EXEC} is the amount of time a user has to wait for a computational task, or a set of tasks, to run from start to finish. For our purposes, ignoring run-time factors such as delays and queueing for available resources, we will assume that for a single CPU task, $t_{EXEC} = t_{TOTCPU} = t_{CPU}$. A multi-CPU task would use more

CPU time than the execution time suggests. Analogously, executing multiple tasks in parallel would demand more t_{TOTCPU} than t_{EXEC} . When the task refers to a search, t_{EXEC} determines the minimum waiting time for a user to finish a search from start to termination, whilst t_{TOTCPU} indicates the computational demand/cost. If a search cannot execute single-CPU tasks concurrently, then $t_{EXEC} = t_{TOTCPU}$. The execution time and the total computational demand of a search are related by the computational (search) intensity. Increasing the search intensity by exploiting concurrent computations shortens its execution time. We will assume that particle simulations are single-CPU tasks.

Another important aspect in measuring the performance of (meta-)heuristics is the *robustness* [129]. We consider a metaheuristic to be robust if it handles the objective function and initial condition well, such that its performance is resistant to poor initial point x_0 . In a similar sense, the search outcome is sensitive to small changes in x_0 if a heuristic is not robust. Moreover, a non-robust heuristic is likelier to have a poorer worst-case scenario than a robust one. In practice, for the purposes of our implementation to the self-assembly problem, we interpret robustness as the dependence of climbing performance on the choice of x_0 —hill-climbing parametrised with high robustness would consistently search the landscape to reach the global peak region regardless of x_0 , whereas the performance of hill-climbing parametrised with low robustness would be heavily dependent on x_0 . We will examine the overall performance of the hill-climbing algorithm as if it were executed in a random initial point by examining several different initial points on \mathcal{S} . Consistent performance across different x_0 is considered as a robust performance.

We consider an algorithm to be *reliable* if, from any given x_0 , it consistently reaches the same basin of attraction, namely the global peak region. We consider an algorithm to have high *accuracy* if it has little variance in the terminal x within the basin of attraction. For example, if an algorithm will always find the global peak region but will return any of, say, 30 points within this region at termination, then it is considered to be reliable but inaccurate. If an algorithm finds the global peak region consistently with most climber terminating at x^* then it is both reliable and robust (e.g. compare Figs. 6.6a and 6.6b on page 101). If an algorithm will statistically only find the global peak region half of the time and be stuck at another local maximum for the other half, but will return only any of, say, 4 points for each respective region, then it is considered unreliable but accurate (e.g. see Fig. 6.7f on page 102).

Computational demand of VMMC simulations and search

Computer simulations of self-assembly are CPU-intensive tasks, especially when modelling large, complex systems at long timescales, and it is under this premise that we consider hill-climbing search as an alternative for parameter tuning. As we saw in Section 4.5.3, our VMMC simulations of 2D patchy particle systems reached up to $t_{EXEC} = t_{TOTCPU} = 1.3 \times 10^8 s$ (5 days), requiring an average of $1.5 \times 10^5 s$ (1.7 days) per simulation per point. With the cost of a single simulation of the order of days, the total CPU time t_{TOTCPU} required to evaluate all 920 points in \mathcal{P} once in a brute-force scan is $1.38 \times 10^8 s = 1552$ days. Parallelism can be trivially exploited in a brute-force scan; if an unlimited number

of cores were available, then all points in \mathcal{P} for our system can be executed simultaneously. The brute-force scan would then be completed within the 5 days it took to perform the longest simulation, although t_{TOTCPU} remains unchanged. Opportunities to exploit such extreme parallelism are limited.

We assume a scan of parameter space with single simulation at each point, as a lower limit for the standard brute-force scan. The peak region may be approximated from a single scan. Averaging over multiple samples at each point is often required to obtain an accurate reading to identify the global peak. This can be chosen at the start—considering more than one simulation (sample) for each point. Efforts to increase the scan accuracy may include user input—progressively contracting the scope of points scanned progressively as more data is collected—in which case a basic scaling of t_{TOTCPU} per full scan with an increase in the number of simulations per points overapproximates the computational demand of the scan.

Such differences in the potential computational performance by varying the distribution of simulation execution is an example of the *trade-off* between competing aspects of the search constraints. The user may not want to wait for $t_{EXEC} = t_{TOTCPU}$ for a search to finish, in which case the exploitation of concurrent simulations is a must to reduce t_{EXEC} . On the other hand, they may be constrained to a finite number of resources or have multiple searches to perform, so a search strategy to reduce t_{TOTCPU} is welcomed. One major disadvantage of hill-climbing is its limited opportunity for exploiting concurrency when executing simulations, since the algorithm is based on iteratively exploring the local landscape, requiring one iteration to finish before another begins. Considering hill-climbing to reduce t_{TOTCPU} becomes relevant when there is a restricted number of CPUs per search (for example, when there is limited concurrency in simulation execution, or when multiple independent searches/systems are considered) such that t_{TOTCPU} is of similar order to t_{EXEC} under the brute-force scheme. We will take the computational demand t_{TOTCPU} to be the default aim for improving the parameter tuning performance. Equivalently, we aim to reduce the number of full duration simulations called by a search, addressing if and when hill-climbing is a more efficient option in comparison to the brute-force scan and exploring different variations of hill-climbing heuristics for doing so.

Approaches to improving performance

There are two approaches to reducing t_{TOTCPU} of the tuning process. The first is to focus on reducing the computational resource of the complex stochastic particle simulations themselves, i.e. reduce t_{TOTCPU} for each particle simulation. As covered in Section 2.4.3, speeding up the particle simulations is the most direct way to improve the computational efficiency of the search.

The second approach is to reduce the number of full duration simulations performed to find our region of interest. We try to address this second approach. We ask if the hill-climbing algorithm can consistently reduce the number of evaluations (and therefore simulations) required to find the optimal or near-optimal region of self-assembly as compared to the brute-force scan. Moreover, as we shall motivate in the next section, we

will attempt to reduce the search intensity (number of evaluations called on average per iteration) to see how it affects the computational demand of a hill climb run.

5.3 Noisy fitness landscape

As is the case for most real world problems, repeated measurements of a self-assembly experiment with the same parameters will yield slightly different results every time. Observables are changing in time, and variability in the final product of a stochastic system is inevitable. These random variations associated with the observations are intrinsic to the system (the physical self-assembly system should also exhibit similar variability), and thus it is important to account for them. We can treat this difference as *noise* about some expected value.

For a solution $x \in \mathcal{S}$, a *noisy evaluation* $g(x)$ can be mathematically represented as:

$$g(x) = f(x) + Y_x, \quad Y_x \sim \mathcal{Y}_x, \quad (5.1)$$

where $f(x)$ represents the *true/underlying fitness value* of x , and Y_x is a *random fluctuation* drawn from some distribution \mathcal{Y}_x , with expected mean of zero such that the $\langle g(x) \rangle = f(x)$. An evaluation g is the arithmetic mean over r independent *samples* g_i i.e. $g = \frac{1}{r} \sum_{i=0}^r g_i$. When $r = 1$, an evaluation is just one sample. Increasing r reduces the noise of $g(x)$ seen by a climber.

For the self-assembly problem, the interaction parameter coordinate pair $\mathbf{x} = (\varepsilon, w) \in \mathcal{P}$ is evaluated by sampling the corresponding patchy colloidal systems. Each sample Q_i is obtained from an independent VMMC simulation measurement. An evaluation Q of size r of a point $\mathbf{x} \in \mathcal{P}$ is the arithmetic mean over r sampled values $Q_i(\mathbf{x}), i = 1, 2, \dots, r$, i.e. $Q(\mathbf{x}) = \frac{1}{r} \sum_{i=0}^r Q_i(\mathbf{x})$.

5.4 Hill-climbing on noisy landscapes

Hill-climbing is one of the simplest algorithms in the family of local search heuristics. With little assumptions about the global state of the problem, it is widely applicable and easy to implement for problems of any size. At each iteration of the algorithm, a climber performing the hill climb attempts to change its state towards increasing fitness values on f until it reaches a peak where no neighbouring states have higher value. It does so by attempting to move to neighbouring points at each iteration, with the protocols involved in move suggestion and move acceptance criterion offering variation in the hill-climbing algorithm [129; 141]. Greedy hill-climbing considers all candidate solutions in the neighbourhood and moves to the point yielding the highest improvement in fitness. First-improvement hill-climbing accepts the first random neighbour that has higher fitness. Stochastic hill-climbing follows the same procedures as first-improvement hill-climbing, but also accepts *neutral moves* (moves between neighbouring points x and x' with $g(x) = g(x')$). Probabilistic hill-climbing introduces a probabilistic acceptance criterion, with the probability of moving proportional to the difference in fitness [142].

Hill-climbing algorithms do not look ahead beyond the next potential step, and often hold information about its current state only, so it requires little memory to run. It is a popular choice when tackling local optimisation problems and can find fairly reasonable solutions in large/infinite state space.

Algorithm 1: Stochastic hill-climbing algorithm on a noisy fitness landscape g .

Input : Initial point x_0 , termination condition

Output: $x_k, g(x_k)$

```

1  $k = 0$ ;
2  $x_k = x_0$ ;
3 while termination condition not satisfied do
4   Evaluate  $g(x_k)$ ;
5   Select random neighbour  $x'_k$ , and evaluate  $g(x'_k)$ ;
6   if  $\Delta g = g(x'_k) - g(x_k) \geq 0$  then
7      $x_{k+1} \leftarrow x'_k$ ;
8   else
9      $x_{k+1} \leftarrow x_k$ ;
10  end
11   $k = k + 1$ ;
12 end
```

Algorithm 1 is a pseudocode of the stochastic hill-climbing algorithm, applicable to a noisy landscape g . At each iteration k , a point x'_k in the neighbourhood of x_k is chosen at random before the climber attempts to move to the point with higher fitness by evaluating both points and comparing their measured fitness, moving to x' if $\Delta g = g(x') - g(x) \geq 0$, but staying at x if $\Delta g < 0$. We call a move attempt a *successful move* if it results in a move to the new location x' , and *unsuccessful* or *failed* if the climber stays at x . During the following iteration $k + 1$, the next pair of points are evaluated even if the same pair of points are considered. The initial point \mathbf{x}_0 can be an arbitrary or informed choice. The former is arguably a careless strategy, as the latter case increases the likelihood of a hill climb reaching global fitness faster. However, initial efforts to derive better \mathbf{x}_0 may require significant time, whereas one can simply ‘execute-and-forget’ a hill climb search if one takes an arbitrary or instinctive approach to setting \mathbf{x}_0 . Again, it is down to the user to judge the initial efforts in deriving better \mathbf{x}_0 , time that can be dedicated to other tasks, against the potential time saved later on in the hill-climbing. Stochastic hill-climbing outperforms greedy hill-climbing on rugged noisy landscapes [141].

The algorithm terminates upon satisfying the termination condition, which can be based on the fitness reached by the algorithm, the allocated number of simulations, or upon reaching a local maximum. A solution x is a local maximum if $g(x) \geq g(x')$ for all $x' \in \mathcal{N}(x)$. Increasing the accuracy of $g(x)$ (by averaging over many samples) is the only way to recognise that x is also a local maximum in f . Compare this to finding the local maxima (and the global maximum) using the brute force method. Multiple measurements per point are required to identify these maxima in f accurately. We did not consider termination conditions based on locating a local maximum, a necessity if the peak value is unknown and requires identification, as our focus is on the climbing behaviour of the

algorithm in general. However, we do make observations on a more tolerant condition, threshold fitness, where a target fitness value is taken to be a sufficient answer and climbs are terminated upon attaining a value equal or greater than this value. This termination condition, requiring the understanding of the measure used to determine a value to target, is relevant in searches where a region of good assembly is desired, but not necessarily the best. This is often the case when searching under a time limit, or when dealing with large scale production where a suitable compromise to the yield is acceptable after considering other aspects of production, e.g. cost of materials, difficulty in operations control, value after production—the order of magnitude, rather than precision, of outcomes is more important.

Hill-climbing often makes quick advances to a solution because of the direct nature of its ascent. However, its design does not allow non-optimal moves to be performed, hence its ability to explore the landscape is heavily dependent on the form of the fitness landscape and the initial point x_0 . Most metaheuristics modify the hill climb to enable the acceptance of non-optimal moves when climbing on noiseless landscapes, for example by explicitly allowing for non-optimal moves in tabu search [131], or using a lag l in late-acceptance hill-climbing [143], where a proposed solution with fitness $g(x'_k)$ is compared with $g(x_{k-l})$. Probabilistic acceptance metaheuristics, e.g. simulated annealing, accept non-optimal moves probabilistically [142].

The presence of noise in the objective function, introducing a soft annealing effect in the landscape, is therefore welcomed in moderation. The smoothing of rough surfaces and the annealing of the peaks enable a local search algorithm to escape a local optimum by allowing a move to be accepted that would not have been otherwise had the noise been absent. Both Levitan [144] and Rana [138] summarised this in their respective local search algorithm studies, suggesting that the presence of small noise enhances the exploration of local search algorithms. However, due to the noisy landscape, many common metaheuristics designed for noiseless landscape would not perform as expected without tailoring. Simulated annealing would only add more uncertainty to the already noisy fitness measurements, while a variant of tabu search which bans climbers from revisiting old points assumes that a previous evaluation is representative of its true fitness. We will expand on the effects of intrinsic noise on the behaviour of stochastic hill climbs in more detail in Sections 5.4.2 and 5.4.3.

5.4.1 Neighbourhood

At each iteration, a hill-climbing algorithm is limited to moving within a local subset of \mathcal{S} defined by the neighbourhood \mathcal{N} . The definition of \mathcal{N} is critical to the performance of any local search algorithms, having large consequences on the behaviour and length of a search algorithm as it explores the fitness landscape [137]. Neighbourhood definitions can be tailored to reflect certain correlations and interaction schemes between attributes that form the parameter space, for example coupling the interactions between different genomes [145].

In colloidal particle design, the design parameters considered form a set of coordi-

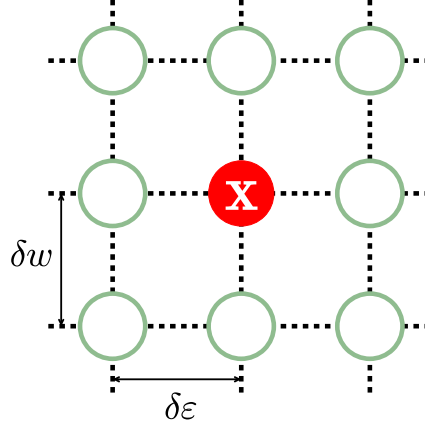


Figure 5.1: The neighbourhood \mathcal{N} definition for the yield landscape on \mathcal{P} , a discretised space of parameter points with spacing $\delta\epsilon$ and δw in the ϵ and w axes respectively. The neighbourhood of a point $\mathbf{x} = (\epsilon, w) \in \mathcal{P}$ (red filled circle) are the 8 possible increment combinations of the parameters (green open circles): $(\epsilon + \delta\epsilon, w)$, $(\epsilon - \delta\epsilon, w)$, $(\epsilon, w + \delta w)$, $(\epsilon, w - \delta w)$, $(\epsilon + \delta\epsilon, w + \delta w)$, $(\epsilon - \delta\epsilon, w - \delta w)$, $(\epsilon + \delta\epsilon, w + \delta w)$ and $(\epsilon + \delta\epsilon, w - \delta w)$.

nate axes and a search space. The naive way of defining the neighbourhood of a set of parameter values is to consider incremental changes to a single parameter such that possible moves are all orthogonal to a parameter axis. Multiple incremental changes in the different values expand the possible directions and routes of exploration for the climber, although the neighbourhood would then include non-uniform step sizes (moves are of different lengths). Identification of the role between different design aspects, such as ratios of coupled interaction parameters and aspect ratio of colloidal rods, can inform more non-trivial neighbourhood definitions to correlate movements on the landscape.

For this study we use the neighbourhood definition illustrated in Figure 5.1. For any point $\mathbf{x} = (\epsilon, w) \in \mathcal{P}$ the neighbourhood is defined as all combinations of incremental changes to the elements in \mathbf{x} that lie within the valid parameter range. Moves orthogonal to the parameter axes have a step size of one incremental unit, whereas the diagonal moves are moves of two incremental units when interpreted as a combination of two orthogonal moves. The neighbourhood size (number of neighbours) of \mathbf{x} varies between 3 at the corners of the search space \mathcal{P} to 8 away from the boundary.

5.4.2 Effects of intrinsic noise

To appreciate the effects of intrinsic noise on the hill-climbing behaviour, let us compare it with the stochastic hill-climbing algorithm on a deterministic (noiseless) landscape. For a deterministic landscape ($g \equiv f$), where a system under the same parameter and conditions returns the same value for all evaluations, accepted moves are always uphill or neutral. Downhill moves are never accepted. A climber gets stuck when it finds a local maximum, ridge, or plateau [146]. When a climber is located at a local maximum, all of its available

neighbours have a fitness value less than $g(x)$. When the climber is on a ridge, unless the grid is discretised such that the ridge lies along one of the neighbourhood move axis, all move attempts are also downhill moves. When on a plateau, only neutral moves ($\Delta g = g(x') - g(x) = 0$) are available. A climber performs a random walk but risks travelling indefinitely if the plateau is large.

Neutral moves can either be accepted or rejected systematically, or with some probability. It is obvious in a flat region where all neighbours are neutral moves that a rejection policy would be fatal to the movement of a climber. Accepting neutral moves can help climbers move out of flat regions and local optima, and stochastic hill-climbing, which indifferently accepts the first improving or neutral moves, are able to reach better solutions than greedy or first-improvement hill-climbing where improving moves are prioritised first [141; 147].

Enabling the acceptance of downhill moves is key to overcoming the problem of getting stuck, and is the essence of the diversification process introduced by the use of metaheuristics. By moving downhill, hill-climbing can move away and out of local maxima and hence the consequence of a poor initial point \mathbf{x}_0 is not fatal. In simulated annealing on a noiseless landscape, a local search proceeds in a similar fashion to a hill-climbing algorithm, but with downhill moves accepted with a probabilistic acceptance scheme according to the fitness difference Δg , typically accepting moves with probability $P(\Delta g, T) = \exp(-\Delta g/T)$ [130]. A global temperature T controls the probabilistic preference of accepting a downhill move and is usually scheduled to vary with the search time. The higher the temperature T , the larger $P(\Delta g, T)$ becomes for accepting non-improving moves, allowing the agent performing the algorithm to move between different local maxima (in physics, metastable states are found at local minima and moving to another metastable state requires moving away from the optima in order to overcome the energy barrier). As T tends towards zero, $P(\Delta g, T)$ must also tend towards zero for $\Delta g < 0$ moves, reducing to the classic deterministic hill climb where only uphill moves are accepted at the zero limit, leading to the eventual restriction of the exploration to a local (and hopefully global) maximum. Simulated annealing is a preferred method for exploring complex (energy) landscapes as it has a better chance of locating the global optimum in a fixed amount of time.

Similarly, for a noisy landscape ($g = f + \mathcal{Y}$) the intrinsic noise enable climbers to perform downhill move. Noise in the fitness measurements effectively blurs out the details of a rough landscapes, annealing the peaks and reducing the chance of climbers getting stuck in local optima. However, the annealing of peaks forces a climber to move indefinitely in the neighbourhood of the global peak once reached and never settle on one solution. Moreover, an improvement in the maximum attainable fitness is only seen when a small amount of noise is present [144]. When noise is significant, the performance of the hill-climbing algorithm (or any other search algorithms) on any function weakens since they trade their strong directional convergence (intensification) towards solution for increased exploration of other sub-optimal points (diversification). The larger the magnitude of the noise, the likelier it is for a climber to walk downhill, until at a large noise limit where a climber cannot discern any of the landscape features and instead performs a random walk.

5.4.3 Details of intrinsic noise effects

Levitan & Kauffmann's study [144] on adaptive walks on NK landscape models reported how noise affects the types of neighbourhood moves that a climber can perform. NK landscape models are theoretical fitness models with vector solutions, used in the study of genomes and biological systems to generalise the complexity of interactions between solutions. Adaptive walks are the local search algorithms performed on such models, with neighbours defined as incremental changes to the elements of the vector solutions. The summary of possible move scenarios they described, which will be covered in this section, is applicable to the case of noisy hill-climbing moves.

Four move scenarios are possible during an iteration of the adaptive walk algorithm. Neutral moves were not accepted. Adopting their notation, let a_1 be the candidate solution currently being considered, and a_2 a neighbour of a_1 . Evaluating these two points using a noisy function g gives $g(a_1)$ and $g(a_2)$. g is of the same form as defined in Equation (5.1).

When $\Delta g = g(a_2) - g(a_1) > 0$ and $\Delta f = f(a_2) - f(a_1) > 0$, then the walk moves to a_2 believing that it is increasing fitness, and indeed it is increasing in the expected fitness (true positive). On the other hand, if $\Delta g > 0$ and $\Delta f \leq 0$, then although the algorithm moves to what it sees as a higher fitness valued point, it is in fact moving downhill in expected fitness (true negative). The other two scenarios cover the remaining two combinations of $\Delta g \leq 0$ and $\Delta f > 0$ (false negative), and $\Delta g \leq 0$ and $\Delta f \leq 0$ (false positive). The relative rates of these moves dictate the performance of a walk.

The preference for fitter solutions (they call this the selection pressure) of an agent performing the adaptive walk forces it uphill towards higher fitness, yet the presence of noise introduces downhill moves, pushing it away from peaks. A survey of the probabilities of true positive and false positive moves, $p(\text{true}+)$ and $p(\text{false}+)$ respectively, showed how this influences the walk dynamics of the algorithm in the long term.

Focusing on the fitness dependence of the probabilities [144],

$$p(\text{true}+) = p(\Delta f > 0 | \Delta g > 0, f) \quad (5.2)$$

and

$$p(\text{false}+) = 1 - p(\text{true}+) = p(\Delta f \leq 0 | \Delta g > 0, f) \quad (5.3)$$

where f is the fitness of a_1 in the move.

Applying Bayes' rule [148] gives

$$p(\text{true}+) = \frac{p(\Delta g > 0 | \Delta f > 0, f)p(\Delta f > 0 | f)}{p(\Delta g > 0 | f)}, \quad (5.4)$$

where the denominator can be further expressed as noisy uphill moves subjected to the probability of uphill and non uphill moves in the underlying landscape:

$$p(\Delta g > 0 | f) = p(\Delta g > 0 | \Delta f > 0, f)p(\Delta f > 0 | f) + p(\Delta g > 0 | \Delta f \leq 0, f)p(\Delta f \leq 0 | f). \quad (5.5)$$

From Equation 5.4, it is immediately apparent that as fitness f increases, a point a_1 will

have fewer neighbours with higher fitness, leading to a decrease in $p(\Delta f > 0|f)$. Hence $p(\text{true}+)$ also decreases with higher f . At the fitness value equivalent to that of the global peak, $p(\Delta f > 0|f) = 0$ and $p(\text{true}+) = 0$. On the other hand, $p(\Delta f > 0) = 1$ and $p(\Delta f \leq 0) = 0$ at a fitness value equivalent to the global minimum, hence $p(\text{true}+) = 1$ as expected. The other move types follow similar probabilities. Moreover, the average difference in f values between neighbours, i.e. $\langle \Delta f \rangle$, determines the influence of noise fluctuations on the move types. If Δf is large compared to the fluctuation, then it is less likely that Δg and Δf will give opposing signs, i.e. $p(\text{true}+)$ is higher. Approaching any peaks sees $\langle \Delta f \rangle$ dropping along with $p(\text{true}+)$ [144].

With $p(\text{true}+) = 1 - p(\text{false}+) = 0$ at the global maximum, and $p(\text{true}+) = 1$ at the global minimum, there exists a fitness value f_x at which $p(\text{true}+) = p(\text{false}+) = 0.5$ —it is equally likely to move uphill and downhill in fitness value. f_x is an attractor of the landscape where walkers will fluctuate about [144]. On average walkers with $f < f_x$ will move to higher fitnesses, and to lower fitnesses when $f > f_x$. This explains why walkers move constantly in a near-optimal region around the peak rather than converge to the peak and stay there. Depending on the form of the noise (such as constant noise throughout the landscape or exponentially increasing/decreasing with fitness), there can even be multiple attractors (but always an odd number) [144].

5.4.4 Reducing noise

With noise intrinsic to many real life problems, noise control motivates many metaheuristic studies. The basis for most noise control studies involves altering the number of samples obtained to inform the decision making of a heuristic. For example, Fitzpatrick [149] was one of the first to compare explicit averaging (averaging over many samples for each evaluation) and implicit averaging (averaging over the population size of the gene pool) in genetic algorithm search. Their research suggested that there are cases where the overall efficiency of genetic algorithms is enhanced more by reducing the sample size in the explicit averaging but increasing the population size in the implicit averaging. Jin & Branke [150] summarised the techniques for handling uncertain environments in evolutionary optimisation.

For hill-climbing algorithms exploring noisy landscapes, only explicit averaging of samples to reduce the measured variance of evaluations is possible. As mentioned previously, for a point x , sampling its fitness r times reduces the error of the mean fitness value by a factor of \sqrt{r} , resulting in a measured fitness $g(x)$ that better estimates the underlying true fitness $f(x)$. Fitness-based decisions thus correlate more with the expected fitness difference Δf . The gain in noise reduction diminishes with growing sample size. Note that the variance of the evaluation is altered, but the variance of the observation (sample) of the system remains unchanged.

Even then, finding the optimal sample size is dependent on the form of the function, both globally and in the neighbourhood. As highlighted by Stonedahl & Stonedahl [151], in noisy landscapes there is a sweet spot for the ideal number of samples to average over, where the search is most efficient to reach a desired fitness level. The total computational

cost of the search is minimal when enough sampling is performed to resolve sufficient detail in the landscape for uphill decisions but not by too much that the evaluation of each point becomes too slow and computationally demanding.

Instead of using a fixed evaluation sample size r throughout the whole search, varying r during the walk may help speed up convergence to optima. One search strategy is to perform a quick exploration at the start of a search to locate basins of attractions before shifting the search algorithm to target finer details in the landscape and converge towards the global peak. This can be achieved by initially starting with low r (large noise) then gradually increasing r —an intensification process of the algorithm—to increase the precision of evaluations. Protocols to increase r include adaptive rules that estimate the current conditions of the search agent, such as the noise level of the current state, or non-adaptive scheduled ones that scale according to the number of iterations. Liu [152] focused on selecting mathematically motivated adaptive and non adaptive rules based on the step size and generation number of the search (each iteration yields a new generation) to maximise the speed of convergence in distance to the solution given a ‘budget’ in the number of evaluations allowed.

Metaheuristics complementing search algorithms externally rather than modifying the search algorithm parameters internally have also been studied. Random restarts, where a climber restarts its climb at a random location every time it becomes stuck at a local maximum or makes little progress, have been found to effectively complement the accurate measuring of fitnesses to enhance the exploration of search space. The fittest solution amongst all these restarted climbs is tracked. The philosophy here is that it is better to abort a climb when it appears hard to traverse locally and start at a potentially better initial point \mathbf{x}_0 , instead of investing more effort into improving the current solution. One way to see this is that one is effectively running several independent hill-climbs at different locations and tracking the best.

When fitness evaluations are computationally expensive, it has been shown that fitness caching, where only newly visited sites are evaluated and stored to build up a picture of the landscape for reuse later, is a possible way to reduce the total computational cost of a search. Stonedahl & Stonedahl [151] applied fitness caching on top of random-restart hill-climbing on noisy fitness landscapes, then combined it with the averaging over multiple samples (they call this noise reduction technique *fitness averaging by repeated sampling*) to reduce the total computational cost of finding peaks. The caching of fitness values as a climber explores a fitness landscape builds up an approximate image of the fitness landscape. The cached landscape is very rough if one sample is taken for the evaluation, and precise for a large evaluation sample size r . In between where r is moderate, the reconstruction will be able to tolerate some level of noise while retaining the peak locations. Additionally, the work attempted to characterise the robustness of several test fitness functions and use different measures from the fitness difference correlation function (the distribution of fitness differences between all neighbouring points x_i, x_j i.e. $|f(x_i) - f(x_j)|$) to select the optimal sample size at which one could still identify the maxima from the noisy image. The experiments demonstrated that an optimal noise level, corresponding to the shortest number of samples required to reach a certain threshold fitness value, exists,

although a universal technique to deduce this noise pre-search via knowledge of the fitness difference correlation function was not found.

5.4.5 Alternative approach—reusing past sampled values

All the examples described above attempt to reduce the total computational cost of a local search algorithm by adjusting the sample size of each evaluation. When each sample is expensive to measure, it is discouraging to increase the number of samples per evaluation. This is the case for our particle simulation based parameter tuning problem, where each simulation is expected to be computationally intensive and expensive already. Increasing the sample size of each evaluation from $r = 1$ to $r = 2$ doubles the effective computational cost t_{TOTCPU} of a climb of length k_{\max} . The computational intensity (the average number of simulations performed per iteration) scales with r . If samples cannot be taken concurrently, then r also affects the execution time t_{EXEC} —each evaluation will take twice as long to complete, doubling the execution time per step. To use these techniques without increasing t_{TOTCPU} or t_{EXEC} requires reducing k_{\max} by half at least. We want to see a ‘return’ in t_{TOTCPU} from raising its computational intensity (an initial ‘investment’).

Noticing this, we ask whether there are alternative ways to control noise without increasing r . Examining the stochastic hill-climbing algorithm defined in Algorithm 1, we observed that $g(x)$ is sampled at every iteration, regardless of the result of the previous step (We use the terms step and iteration interchangeably). More specifically, the same current location is reevaluated following a failed move in the $(k - 1)$ th iteration, even though $g(x_{k-1})$ is still a valid representation of $x_k \equiv x_{k-1}$. We ask whether it is necessary to perform this reevaluation after every failed attempt. We introduce a concept of a trial where a current fitness value $g(x)$ is reused for multiple iterations. The immediate local consequence in reusing $g(x)$ that the computational intensity of these iterations is halved—only one evaluation is called rather than two. A decrease in computational intensity would see a climb of equal length k_{\max} demand fewer computational resources, producing a direct scaling down of t_{TOTCPU} without having to invest in more computational resources initially. Another search protocol we will study is a cumulative averaging protocol to average multiple samples across past iterations rather than averaging multiple samples within each single iteration.

For the second objective of this thesis, we seek to answer whether reducing the average number of evaluations performed per iteration via the reuse of past values can reduce t_{EXEC} and t_{TOTCPU} of a stochastic hill-climbing heuristic, while increasing the yield of the terminal solutions reached. In the next chapter, we will relax the requirements of the heuristic for re-evaluating the current point’s fitness value at every step, instead allowing an evaluation to be used over m multiple iterations. Other search strategies to be considered will also avoid increasing the computational intensity of the search. The averaging of fitness values is crucial to tackle the annealing of peaks, so that a climber can eventually converge to a local maximum. We will consider a cumulative averaging of fitness values over past evaluations, which requires no extra simulations per evaluation, as an alternative to fitness averaging over several samples within each evaluation.

5.5 Conclusion

In this chapter, a framework borrowed from the study of optimisation problems was presented for studying the self-assembly tuning problem. A noisy fitness landscape was defined to model the self-assembly yield landscape, on which the hill-climbing heuristic can be applied. Heuristics are approximate algorithms that work under a set of constraints to find optimal or near-optimal solutions of a fitness landscape within practical times. Constraints relevant to simulation-based self-assembly parameter tuning were identified: simulations of particle systems take a long time to run and are computationally demanding; and the self-assembly observations are noisy. We motivated our study of hill-climbing as an alternative methodology to the brute-force scan for finding the best yielding parameter values more efficiently. We will empirically test this in the next chapter. A possible metaheuristic, focusing on reducing the intensity of the hill-climbing algorithm by the use of trials to improve the efficiency of the stochastic hill-climbing efficiency was proposed and will be formally defined and studied in the next chapter along with other ‘investment-free’ strategies: multi-duration climbing schedule and cumulative averaging.

Chapter 6

Hill-climbing on self-assembly landscapes

Our analysis of the hill-climbing heuristics is based on how well it guides the execution of MC simulations of a colloidal particle system for tuning an identified set of parameters. The main characteristics of the yield landscape is a large noisy yield where measurements have large fluctuations, along with expensive and time consuming simulations required to obtain the associated yield values of the system of interest. Computational factors provide several limitations, such as the size and complexity of the system and the available resources, which determine how long a simulation will take to run and therefore limit the practicalities of the search heuristic. As we will see, it also naturally restricts our ability to study hill-climbing whilst performing large numbers of simulations. One needs large samples of hill climbs in order to draw any reliable conclusions on the performance from the statistics.

To make progress, we constructed a toy model \hat{Q} to generate expected yield values for given parameter values. We performed hill climbs on this toy model to examine the expected performance of hill-climbing on the actual VMMC simulation setup. Based on the mean and variance of the actual VMMC simulation data from Chapter 4, \hat{Q} produces similar noisy outputs to the actual VMMC simulations, agreeing especially well for high Q values. Since uncertainty and noise is intrinsic to the measurements in the actual self-assembly system, we expect the small differences between \hat{Q} and VMMC simulations to have little effect on the qualitative behaviour of the hill-climbing runs. This model also enabled us to test several hill-climbing variants in order to suggest new possible metaheuristic tools to complement search strategies involving costly evaluations of particle systems.

As we learned from Chapters 4 and 5, yield landscapes of self-assembly systems are typically noisy, with a localised region of high yielding assembly with time-dependent optima. Common metaheuristics on noisy landscapes focus on controlling the noise level by *increasing* the sample size of each evaluation, which is undesirable when each evaluation is expensive and time consuming to perform. We seek to find strategies and heuristics that extend the standard stochastic hill-climbing algorithm in order to reduce the computational resources required. Our main contribution focuses on a new protocol where a

current fitness value is reused for m consecutive iterations. Aside from testing the potential of hill-climbing to replace brute-force scan, we empirically test metaheuristics to address the following questions:

- Can one reduce the number of Monte Carlo simulations executed in a hill-climbing search whilst maintaining desirable algorithm traits and outcomes including reliable and high terminal yield (synonymous with good yielding design parameters), and robustness to poor initial points?
- What aspects of noisy yield landscapes can be exploited to improve the terminal yield, robustness and reliability of hill-climbing?

\hat{Q} serves as an example parametrised self-assembly tuning problem (parametrised here refers to the optimisation parameters and step size choices). The choice of state points used determines the local maxima of the yield landscape that our experiments will be performed on, thus the numerical results are subject to \mathcal{P} and \hat{Q} . Our focus is therefore to learn about the qualitative results that should remain valid across other self-assembly systems under different parametrisations, with emphasis placed on how metaheuristic protocols alter the reliability and robustness of hill-climbing. This serves to guide any potential users of their optimisation parameter choices and adjustments according to their optimisation problem. Hill-climbing performance will be subject to the initial point \mathbf{x}_0 —good \mathbf{x}_0 choices will improve the search outcomes. We will consider the search performance when hill-climbing is initialised with random \mathbf{x}_0 choices, in order to provide the baseline for exploring the worst-case scenario, although we will only present a select number of initial points to provide clarity and comparison between different \mathbf{x}_0 . Indeed, in practice, carefully considered \mathbf{x}_0 choices will offer better conditions for the success of a hill-climbing run in finding high yielding parameters at relative cost but this is problem-specific and beyond the objectives of this thesis.

In the following section a preliminary study of hill-climbing for parameter tuning is given, where an actual application of hill-climbing alongside MC simulations were performed to illustrate its potential. The limited conclusions drawn due to the small set of climbs performed motivates our decision to move on to studying the hill-climbing on a model landscape to replace the running of full VMMC simulations. This yield landscape model is constructed in Section 6.2 using the data gathered in Chapter 4. Section 6.3 presents the stochastic hill-climbing modified by the trial protocol, along with cumulative averaging and other protocols studied. The behaviour of these algorithms on our yield landscape model is discussed in Section 6.4. Next, the analysis of move probabilities covered in Section 5.4.2 is extended to account for the effects of the trial period limit m . Following this, threshold hill-climbing experiments are performed, both on the yield landscape model and on 1D model fitness landscapes to identify the optimal m^* that reduces the total number of evaluations required for a climber to reach a certain threshold fitness. Finally, the climbing schedule strategy utilised in the preliminary studies is revisited and studied formally on our yield landscape model with the modified hill-climbing algorithm.

6.1 Pre-model studies: direct application of hill-climbing with VMMC simulations

Before moving on to looking at the toy yield landscape model for studying hill climbs, we had directly deployed a hill-climbing algorithm along with the VMMC simulations to allow the algorithm to determine the sequence of parameters executed by the computer in order to locate the best yielding parameters, as a preliminary check on the feasibility of the approach.

Specifically, we used the greedy hill climb (Algorithm 2) for the preliminary studies, exploiting the computational resources available to us to simultaneously evaluate the current point and all neighbouring points. Although conclusions about the hill-climbing performance are limited due to early termination of these direct climbs and the small number of climbs performed, we demonstrated a few empirical strategies that are worth mentioning. These strategies exploit the time-dependency of dynamical self-assembly yield landscapes.

6.1.1 Direct application of greedy hill-climbing algorithm with VMMC simulations to tune parameters

The pseudocode for the greedy hill-climbing algorithm is given in Algorithm 2. At each iteration k , nine simulations are executed—one for each evaluation of the current climber position \mathbf{x}_k and its neighbouring state points $\mathbf{x}'_{k,j}$ in $\mathcal{N}(\mathbf{x})$ —to give a representative snapshot of the local neighbourhood landscape. The neighbouring fitnesses $g(\mathbf{x}'_{k,j})$ are ranked and the climber moves to the $\mathbf{x}'_{k,j}$ with the highest fitness $g(\mathbf{x}'_{k,j})$ if it is larger than $g(\mathbf{x}_k)$, else it stays at \mathbf{x}_k . When two or more neighbouring points share the same highest fitness value the climber moves to one of these points with equal probability.

A shorter simulation duration of $t_f = 10^7$ VMMC cycles was opted in this preliminary study to limit the execution time of a search. The rest of the simulation setup was kept the same as in Section 4.2 ($N = 900$, $\eta = 0.04$, square box, $k_B = 1$, $T = 1$, initialised in a random well-mixed configuration).

A *climbing schedule*—defined by a fixed duration update size Δt_V (in units of VMMC cycles) and an update rate κ —was used to provide a termination condition and further reduce the potential execution time. At the start of this climbing schedule, simulations were performed for $t_V = \Delta t_V$. t_V was then increased by Δt_V every κ iterations until the simulation duration reached $t_V = t_f$, where simulations of length t_f were performed for another κ iterations before the climb was terminated. A tolerance trigger Q_{\min} was used to control the start of this climbing schedule. Prior to the start of the climbing schedule, all simulations were limited to a simulation duration of $t_V = \Delta t_V/2$. Upon measuring a simulation with value $Q > Q_{\min}$, the climbing schedule was triggered.

The idea behind introducing these additional protocols is to prevent climbers from launching full length simulations right from the very beginning, when they are likelier to be in a poor yielding region of the landscape. Since Q of a system generally increases as it evolves (see Section 4.5.1), a system with a significant Q value at early times will be expected to perform at least as well during late times; it is unlikely to have low Q at

Algorithm 2: Greedy hill-climbing algorithm with neutral moves accepted probabilistically.

Input : Initial point x_0 , termination condition.
Output: $x_k, g(x_k)$

```

1  $k = 0$ ;
2  $x_k = x_0$ ;
3 while termination condition not satisfied do
4   Evaluate  $g(x_k)$ ;
5   for every neighbour  $x'_k$  do
6     Evaluate  $g(x'_k)$ ;
7   end
8    $g_{\max} = \max\{g(x'_k) \text{ for all neighbours } x'_k\}$  ;
9    $\Delta g = g_{\max} - g(x_k)$ ;
10  if  $\Delta g > 0$  then
11     $x_{k+1} \leftarrow x'_k$  ;
12  else if  $\Delta g = 0$  then
13    Move to  $x_k$  or  $x'_k$  with equal probability;
14  else
15     $x_{k+1} \leftarrow x_k$  ;
16  end
17   $k = k + 1$ ;
18 end
```

final time t_f . Following a similar reasoning, a system with low Q at t_f will have low Q for all $0 \leq t_V \leq t_f$. Therefore it is wiser to keep the simulation duration low, searching for indications of possible self-assembly activity at early times, while moving away from regions likelier to lead to poor yield. Arriving at a potentially high yielding solution by simulating only a fraction of t_f reduces the computational cost and execution time of the hill-climbing searches. The trigger keeps the initial simulation duration low until a desired tolerance Q_{\min} of self-assembly activity is detected in the neighbourhood, in case climbers are initiated too far from the hill, e.g. in the fluid phase region, prompting a climb to enter its climbing schedule phase at more probable regions of high yield.

We chose to use $\Delta t_V = 2 \times 10^6$ VMMC cycles and $\kappa = 5$ iterations for our hill-climbing runs to make sure the hill-climbing run terminated within reasonable times. We chose to use $Q_{\min} = 0$ since the shape factor in the definition of Q ensures that only clusters with 6 or more particles have a non-zero Q . Under these conditions, the quickest a climber would be able to terminate the climb is after 26 iterations, enabling climbers to travel a maximum of half the range in both ε and w axes within this shortest climb length.

6.1.2 Observations

One such example climb is shown in Figure 6.1 on page 86. Initialised at $\mathbf{x}_0 = (12, 12)$ away from the hill, where low yielding polygonal structures are found, the climber steadily moved towards lower ε and w onto the hill. The trajectory is roughly split into three stages based on Q . The first stage occurred for low Q before the trigger was activated and at the early stage of the climbing schedule. In this example, the climbing schedule

was triggered right from the start: $t_V = \Delta t_V = 2 \times 10^6$ VMMC cycles at $k = 1$, and continued for $1 \leq k \leq 5$ as it explored the low yielding region. The second stage occurred from $t_V = 4 \times 10^6$ VMMC cycles. This main climbing stage, defined by a large increase in Q , lasted for 3 iterations only as the climber located at the foot of the hill climbs to significantly fitter neighbours. This stage is highly directional, being a combination of increase in time and climbing of the hill. After the rapid climb to $Q > 300$, the climber entered the final stage of the climb as the climbing schedule increased to t_f , exploring the neighbourhood of the peak region with Q fluctuating largely between 300 and 500 until the termination condition is reached. Higher activity in ε than in w is observed due to the finer discretisation in ε yielding smaller fitness changes.

The paths of five climbs (including the above example) are shown in Figure 6.2 on top of an interpolated version of the final time yield landscape. Most climbs initialised from different parts of parameter space found the peak region before termination. At this discretisation, the climber starting in the ridge at $(14, 2)$ struggled to move towards lower ε , instead being trapped at higher ε . We were only able to perform a low number of climbs within a reasonable amount of time, hence observations of reliable convergence of climbs are not conclusive. The hill-climbing runs performed in this preliminary study suggests it is a feasible alternative to brute-force scanning. To avoid spending most of the computational time of the search on the VMMC simulations, we decided to move away from the full simulation searches, instead constructing a toy yield landscape model, introduced in the next section, on which to perform the experiments to gain more statistics.

Apart from this, our preliminary studies prompted us to base our study on the stochastic hill-climbing algorithm (Algorithm 1) in the rest of this work. With the greedy hill-climbing algorithm, it is computationally wasteful to decide the best neighbour to move to after sampling a full picture of a local neighbourhood, whose representation of the underlying fitness landscape is poor due to fluctuations. Moreover, stochastic hill-climbing has higher explorative power on rugged landscapes than greedy hill-climbing, reaching fitter solutions more consistently [141]. Instead, redistributing the computational intensity (resources required per evaluation) in greedy hill-climbing to perform multiple stochastic hill-climbing runs is likely to increase the reliability of the results collectively, or at least increase the chance of locating the global peak region if different \mathbf{x}_0 are used. For example, for the same computational intensity required to evaluate all eight neighbouring points and the current point per iteration in a greedy hill climb for the neighbour defined in Section 5.4.1 (9 evaluations per iteration), four independent stochastic hill climbs—each requiring two evaluations per iteration—can be executed, massively increasing the chance of successfully finding the global optimal solutions.

6.2 Yield landscape model \hat{Q}

6.2.1 Yield landscape model construction

The interaction between the search algorithms studied and the VMMC simulations of self-assembly systems are rather straightforward. The algorithm provides inputs for the

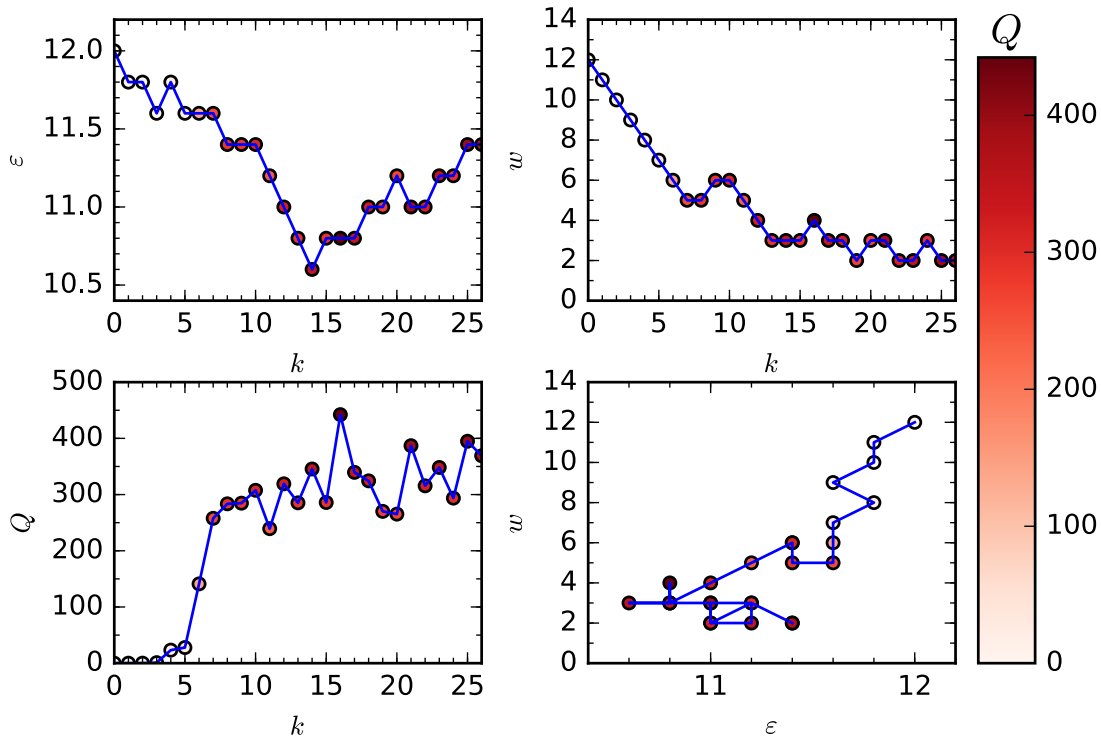


Figure 6.1: Example trajectory of a hill climb search in the parameter space of patchy particle model. The climber performing the search starts at $\mathbf{x}_0 = (12.0, 12.0)$. Clockwise from top left: changes in the climber's ε over k ; changes in the climber's w over k ; the path of the climber on the (ε, w) parameter space; and the maximum evaluated yield value $\max\{Q(\mathbf{x}_k), Q(\mathbf{x}'_k)\}$ during the iteration k —the value is also depicted in the colour gradient of the symbol fill in all the plots (a darker shade (red) corresponds to a larger value).

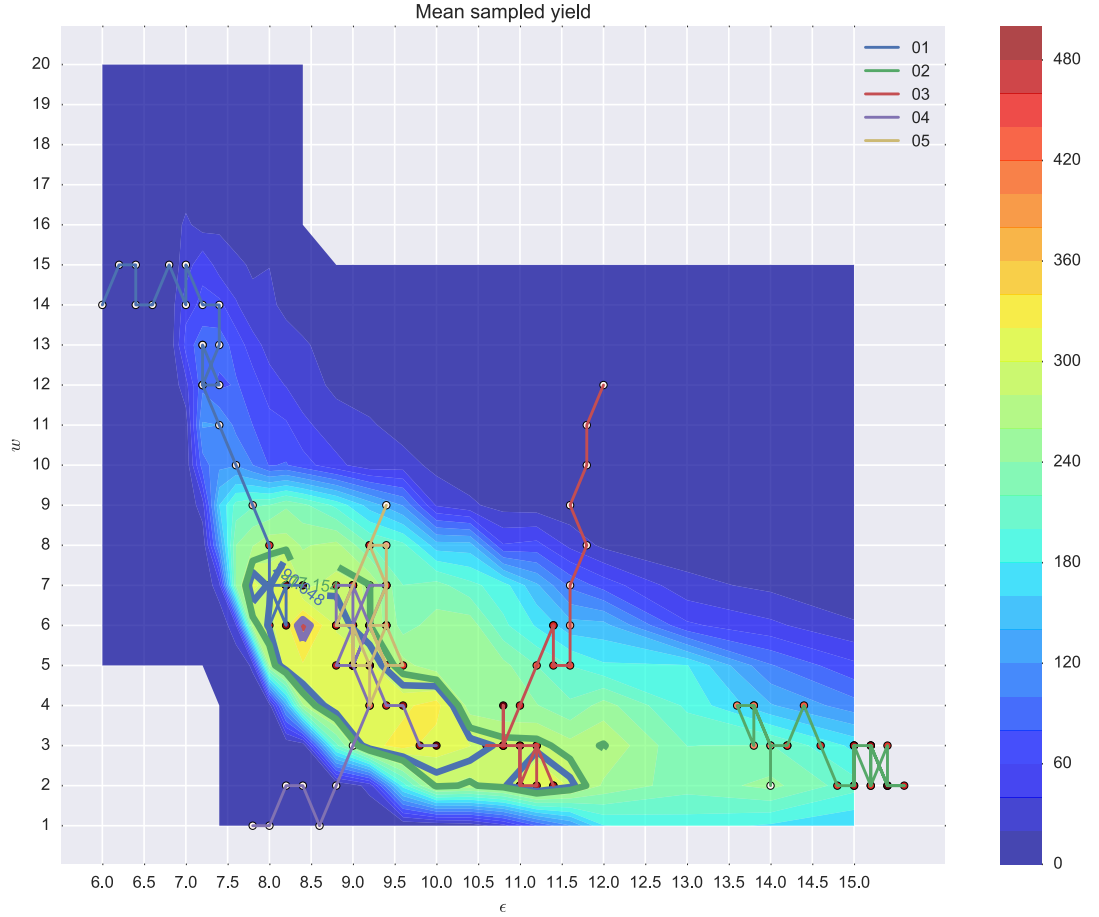


Figure 6.2: The path of five greedy hill-climbing runs coupled with VMMC simulations are shown on an linearly interpolated version of the yield landscape data. The initial points are $(6, 14)$, $(14, 2)$, $(12, 12)$, $(7.8, 1)$, $(9.4, 9)$. Climbers starting away from the hill have an initial fitness close to zero, but all managed to reach a terminal fitness $f > 200$ from different directions. The climber starting at $(14.0, 2.0)$ around the ridge fails to escape towards higher fitness at lower ε within the short climb time, instead being ‘trapped’ at higher ε than it started with.

VMMC simulations, which in turn returns a yield value to inform the move protocol of the local move landscape from which a decision is made. We constructed a yield landscape model as a substitute for the VMMC simulations to bypass the large computational effort required. The role of the yield landscape model, essentially an array of random number generators each associated with a distinct state point \mathbf{x} in parameter space, is to provide noisy outputs resembling the act of performing the actual VMMC simulation itself at the corresponding \mathbf{x} . We aim to capture the broad features of the yield landscape, namely the overall form of the hill, the sharp front at the fluid to fluid-solid dynamical phase boundary, and the \mathbf{x} -dependent yield fluctuations.

We based our yield landscape model on the available data from the brute-force scan of the parameter space in Section 4.4.1 (see Fig. 4.8 on page 48). Q is bounded by the interval $[0, N]$, where N is the size of the system, hence a truncated normal distribution (TND) (also known as truncated Gaussian distribution) was used to approximate the noise contributions involved in measuring Q . From the noisy landscape definition in Equation (5.1), at each point $\mathbf{x} = (\varepsilon, w)$ sampled in the brute-force scan, the measured model yield $\hat{Q}(\mathbf{x})$ is defined by

$$\hat{Q}(\mathbf{x}) = \langle Q(\mathbf{x}) \rangle + Y_{\mathbf{x}}, \quad Y_{\mathbf{x}} \sim \mathcal{Y}(0, \sigma(\mathbf{x}), a, b), \quad (6.1)$$

where $\langle Q(\mathbf{x}) \rangle$ and $\sigma(\mathbf{x})$ are the mean and standard deviation of the simulation yield data at \mathbf{x} , and a random variable $Y_{\mathbf{x}}$ is drawn from a truncated normal distribution $\mathcal{Y}(0, \sigma(\mathbf{x}), a, b)$. $\mathcal{Y}(0, \sigma(\mathbf{x}), a, b)$ is constructed by truncating a normal distribution $\mathcal{N}_{\text{norm}}(0, \sigma(\mathbf{x}))$ of zero mean and standard deviation σ by a lower bound $a = -\langle Q(\mathbf{x}) \rangle$ and upper bound $b = N - \langle Q(\mathbf{x}) \rangle$, such that $0 \leq Q \leq N$ for all random terms $Y \sim \mathcal{Y}$. We discuss the use of the TND as the noise distribution, and the similarity between the resulting model and the simulation data in the next section.

6.2.2 Comparison of the yield landscape model \hat{Q} with simulation yield data Q

The form of the yield value fluctuation distribution is expected to be dependent on the state point \mathbf{x} . For example, we expect a binomial distribution at the boundary of fluid and fluid-solid coexistence, where the self-assembly outcome depends heavily on whether nucleation events are observed. This is different to higher ε regions where the central limit theorem is expected to be a sufficient approximation of the yield outcomes. With between 10 to 16 yield values measured per state point \mathbf{x} , we could not resolve the details of any distribution with any confidence, hence the TND is used throughout. Moreover, to avoid overfitting the TND to match its mean and variance with the data values, the normal distribution was fitted to the statistical moments of the data before it was truncated accordingly to give the TND. The noisy outputs of this model hence does not reproduce any skewed or binomial distribution, or outliers that is otherwise possible in VMMC simulation steps, which is expected to have an effect on the local behaviour of the hill-climbing algorithm. However, this concerns only a fraction of the state points considered, whose TND is subject to large σ , hence we believe the qualitative behaviour of the hill-climbing observations on \hat{Q} will accurately reflect those performed with actual VMMC simulations.

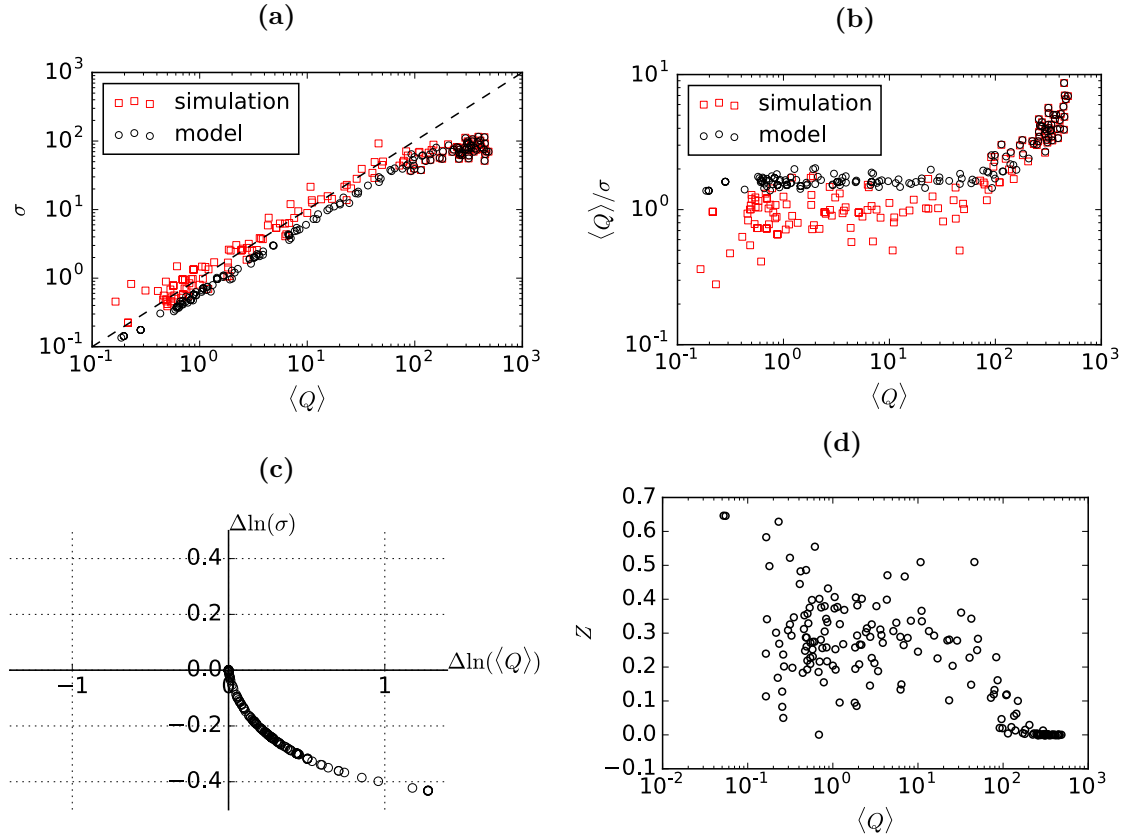


Figure 6.3: Comparison of the yield landscape model and the simulation data. (a) The standard deviation $\sigma = \sqrt{\text{Var}(Q)}$ is plotted against $\langle Q \rangle$ on log-log scale for the simulation data and the yield landscape model, obtained by taking 16 samples per point. σ scales linearly with $\langle Q \rangle$ in both cases, following the dashed line $\sigma = \langle Q \rangle$, but deviates from this for $\langle Q \rangle > 50$ (b) The signal-to-noise ratio $\langle Q \rangle / \sigma$ agrees with this, staying relatively constant between 1 and 2 for $\langle Q \rangle < 50$, before increasing for $\langle Q \rangle > 50$. (c) The differences in $\ln(\sigma)$ and $\ln(\langle Q \rangle)$ between the model values and the corresponding simulation values are no larger than 1 in each axis, with points where the model mean fitness is larger also having lower noise (data points populate the bottom right quadrant of the plot). Plotting the magnitude of these differences $|(\Delta \ln(\langle Q \rangle), \Delta \ln(\sigma))|$ against $\langle Q \rangle$ shows that the larger deviations of \hat{Q} from simulated Q happens for small $\langle \hat{Q} \rangle$, analogous to (d) the z -score plotted against $\langle Q \rangle$.

The distribution of $\langle \hat{Q} \rangle$ and $\langle Q \rangle$ is found to be very similar. The σ and $\langle Q \rangle$ scatter plot in Figure 6.3a exhibits a positive linear correlation, a trend also captured by $\langle \hat{Q} \rangle$. The truncation redistributes the probability mass over the resulting domain, resulting in a shifted expectation value of \mathcal{Y} and a shrunken standard deviation, i.e. $\langle \hat{Q} \rangle \neq \langle Q \rangle$ and $\hat{\sigma} < \sigma$, as shown in Figs. 6.3c and 6.3a. The *signal-to-noise* ratio in Figure 6.3b is constant for $\langle Q \rangle \lesssim 50$, reflecting the additive contributions from the particle components of the clusters, but increases for $\langle Q \rangle > 50$ by up to an order of magnitude more. This is likely a reflection of how state points exhibiting high yield are also more consistent at achieving such values; there is high variability in the low yielding structures. $\langle \hat{Q} \rangle$ exhibits a slightly stronger signal compared to the simulation, suggesting that climbers on \hat{Q} do not feel as much fluctuations at low yield value, but is of the same form.

Further comparison between the model and simulation data is performed by studying $Q(\mathbf{x})$ and $\langle Q(\mathbf{x}) \rangle$ for all \mathbf{x} . In Fig. 6.3c, $\Delta \ln(\langle Q \rangle(\mathbf{x})) = \ln(\langle \hat{Q}(\mathbf{x}) \rangle) - \ln(\langle Q(\mathbf{x}) \rangle)$ is positive for almost all sampled values since $\langle Q \rangle(\mathbf{x}) < N/2$ for most points \mathbf{x} . The differences in the logged σ values, $\Delta \ln(\sigma(\mathbf{x})) = \ln(\hat{\sigma}(\mathbf{x})) - \ln(\sigma(\mathbf{x}))$, are also limited in order of magnitude. The correlation in Fig. 6.3c is due to the same boundaries being applied across all truncations.

Moreover, the magnitude of $(\Delta \ln(\langle Q \rangle), \Delta \ln(\sigma))$ plotted against $\langle Q \rangle$ exhibits the same distribution as the z -score plotted against $\langle Q \rangle$ (Figure 6.3d). The z -score [153] (also known as standardised score) compares a sample with mean X_1 to a distribution with known mean μ and standard deviation σ in units of the standard deviation σ , assuming a normal distribution:

$$z = \frac{X_1 - \mu}{\sigma^2}. \quad (6.2)$$

This gives an idea of how well the sample mean of each TND approximates the normal distributions that used the simulation data, with z closer to zero for distributions that are more similar to each other.. From studying Fig. 6.3d, the model output agrees very well for points with $\langle Q \rangle > 200$ as the mean of the distribution is sufficiently far enough from the $[0, N]$ boundary. For example at \mathbf{x}^* , rounded to 2 decimal places, $\langle Q \rangle = 522.73$ with $\sigma = 54.28$ and $\langle \hat{Q} \rangle = 522.77$ with $\hat{\sigma} = 54.24$. Points with $\hat{Q} \leq 200$ show less agreement in values, due in part to the lower truncation of 0 being much closer to the mean μ , thus more probability mass is redistributed.

We note that the relationship between the fitness and noise was shown to have an effect on the local search behaviour [144]. The expected simulation values $\langle Q \rangle$ approximates the true fitness $\bar{Q} = \langle \hat{Q} \rangle$ of the yield landscape model \hat{Q} with similar trends. We felt that larger variations for points with low Q is acceptable given its absolute magnitude compared to the rest of the hill in addition to the relatively low precision of these points due to the smaller sample sizes of the simulation data. Considering these small differences between the model and data, we assume the true yield value of \hat{Q} to be $\bar{Q} := \langle Q \rangle$ with standard deviation $\hat{\sigma} := \sigma$ for the rest of study.

6.3 Hill-climbing algorithms

Equipped with the yield landscape model, we now consider the hill-climbing protocols used to modify the stochastic hill-climbing algorithm (Algorithm 1). This includes a new parameter m modifying the trial period limit (see below), an aspect that we believe has not been explored yet, and cumulative averaging of past evaluations from previous visits. We consider the neighbourhood selection protocol as well, a technical detail necessary to complete the algorithm.

We did not consider random restarts in our hill-climbing studies. Our trial period protocol is based on detecting the lack of progress when current fitness values are reused. Random restarts also use the lack of progress to determine if a climb should be restarted at a new random x_0 . Since they respond to similar conditions, a distinction is required to make sure the conditions at which these two routines are triggered do not overlap. We leave this as future work.

6.3.1 Trial period limit m

As mentioned in Section 5.4.5, the reevaluation of the current fitness value $g(x)$ at every iteration felt unnecessary. To address this we modified Algorithm 1 by introducing an extra parameter m to limit the number of times an evaluated fitness value $g(x_k)$ is reused giving Algorithm 3. Should a climber fail to successfully move after $g(x_k)$ and $g(x'_k)$ are evaluated, then in the following iteration $k + 1$, a random neighbour x'_{k+1} is again chosen and evaluated, whereas $g(x_k)$ is reused i.e. $g(x_{k+1}) = g(x_k)$. We define a *trial* to be a sequence of iterations where the same evaluation $g(x)$ is used. The number of iterations in a trial is the *trial period*. The *trial period limit* m of the climb is the maximum trial period allowed for any trial during the climb: $g(x_{k+l}) = g(x_k)$ for $0 \leq l < m$ as long as $x_{k+l} = x_{k+l-1} = \dots = x_{k+1} = x_k$. If by step $k + m$ the climber has yet to move away from x_k (i.e. $x_{k+m} \equiv x_k$) then x_{k+m} is evaluated to yield a new evaluated current fitness $g(x_{k+m})$. A new trial also starts upon a successful move. For the next section, the termination condition is a maximum climbing length k_{\max} . All cases where the trial period limit is greater than the climbing length, i.e. $m > k_{\max}$, are degenerate. When $m = 1$, Algorithm 3 reduces to Algorithm 1.

6.3.2 Neighbour selection scheme

A technical detail when implementing the stochastic hill-climbing algorithm concerns choosing whether the evaluation of the neighbourhood is exhaustive (selecting neighbours without replacement), and whether the order of an exhaustive neighbour selection is systematic or randomised. An empirical study by Basseur and co.workers [141] suggested that for a small finite neighbourhood on a noiseless landscape, an exhaustive neighbourhood evaluation is preferable to a random one, while a random neighbourhood exploration order is more natural considering the motivation for randomness by choosing to use a stochastic search. Their experiments did not see significant differences between the different choices however.

Algorithm 3: Modified stochastic hill-climbing algorithm with trial period limit m , on a noisy fitness landscape g .

Input : Initial point x_0 , trial period limit m , termination condition, the maximum number of steps k_{\max} , and m , the number of move attempts/steps per $g(x)$.

Output: $x_k, g(x_k)$

```

1  $j = k = 0$ ;
2  $x_k = x_0$ ;
3 while termination condition not satisfied do
4   if  $j == 0$  then
5     Evaluate  $g(x_k)$ ;
6   end
7   Select random neighbour  $x'_k$ , and evaluate  $g(x'_k)$ ;
8   if  $\Delta g = g(x'_k) - g(x_k) \geq 0$  then
9      $x_{k+1} \leftarrow x'_k$ ;
10     $j = 0$ ;
11  else
12     $x_{k+1} \leftarrow x'_k$ ;
13     $j = (j + 1) \bmod m$ ;
14  end
15   $k = k + 1$ ;
16 end
```

In the following experiment, to observe whether the terminal noisy fitness is influenced by such neighbourhood selection strategies, an exhaustive and a non-exhaustive version of a random neighbourhood exploration order will be considered. Randomly chosen neighbours are not replaced in the move pool for the following iteration in the former version, denoted protocol **a**, until the move pool is exhausted and refreshed, but are immediately replaced in the latter version, denoted protocol **b**.

6.3.3 Cumulative averaging of past sampled values

We considered averaging fitness values of a point across multiple visits rather than discarding the measurements immediately after use. We will determine empirically whether this helps climbers converge to a more localised region of space and reach higher terminal fitness.

For each point $x \in \mathcal{S}$, upon evaluation a single sample $g_j(x)$ is obtained and appended to a history of sampled values $\mathbf{history}(x) = \{g_j(x)\}$. Then, a measured fitness evaluation $g(x)$ is returned by taking the mean of all the samples in $\mathbf{history}$, as shown in Algorithm 4. This approach progressively reduces in the fluctuation seen as the climber evaluates x repeatedly, scaling with the inverse of the size of $\mathbf{history}(x)$, and is expected to add intensification to the algorithm without increasing the sample size of the standard fitness evaluations. This differs from progressively increasing the sample size r of the number of samples to obtain in a single iteration, instead requiring r visits in order to reach a history of r samples.

Each $\mathbf{history}(x)$ is limited in memory by $r_{\mathbf{hist}}$ such that a new sample replaces the

oldest sample in $\text{history}(x)$ when r_{hist} samples have already stored. The choice for r_{hist} could be down to a limit in computational memory, or to limit the noise reduction so that climbers still experience some fluctuations to help drive exploration. The use of $\text{history}(x)$ introduces correlations in $g(x)$ evaluations. If all samples are stored and averaged throughout the search, then in practice $\text{history}(x)$ can be replaced with two numbers tracking the averaged fitness value and the number of samples considered.

Algorithm 4: Cumulative averaging metaheuristic for noisy hill-climbing evaluation at x .

Input : Point x , $\text{history}(x)$ array tracking past sampled values, maximum number of samples r_{hist} .

Output: $g(x)$

```

1 if  $\text{length}(\text{history}(x)) \geq r_{\text{hist}}$  then
2   | Delete oldest entry in  $\text{history}$  ;
3 end
4 Sample  $x$  to obtain  $g_{\text{sample}}(x)$ ;
5 Add  $g_{\text{sample}}(x)$  to  $\text{history}(x)$ ;
6 Return  $g(x) = \frac{1}{\text{length}(\text{history}(x))} \sum_j g_j$  for  $g_j$  in  $\text{history}(x)$ ;
```

6.4 Hill-climbing observations

In this section we present observations from executing Algorithm 3 augmented by different protocols on the yield landscape model \hat{Q} . We will describe the effects of m on the hill-climbing behaviour in terms of the measured and true fitness of the climbers, for individual climbing runs and for a collective set of runs to determine the expected average behaviour of a hill-climbing protocol. A total of 24 different hill-climbing variants were considered by combining the different protocols mentioned in the previous section, including the trial period limits $m = 1, 8, 1000$ (denoted by protocol names **m1**, **m8** and **m1000**) and with no retrials of the current fitness (**NR**); non-exhaustive (**a**) and exhaustive (**b**) random neighbour selection; and cumulative averaging with sample sizes $r_{\text{hist}} = 1$ (**S1**), $r_{\text{hist}} = 16$ (**S16**) and averaging over all past samples (**WA** i.e. with averaging). We include the protocol **NR** that is adopted by climbers on a deterministic landscape where the current fitness $\hat{Q}_k(\mathbf{x})$ is taken to be the higher of the two fitnesses in the previous iteration i.e. $\hat{Q}_k = \max\{\hat{Q}_{k-1}(\mathbf{x}), \hat{Q}_{k-1}(\mathbf{x}')\}$, to demonstrate the necessity of reevaluating the current fitnesses upon a successful move on a noisy landscape. A **NR** climber will therefore always have a non-decreasing \hat{Q} during the run. Climbs were terminated after $k_{\text{max}} = 1000$ iterations.

The hill-climbing outcome was highly dependent on the initial point. This is system dependent—the same protocols and search parameters will produce slightly different outcomes depending on the properties of the noisy fitness landscape. What we are interested in is the overall effect of the behaviour and how different choices in parameter values may influence the robustness and reliability of the search and its outcome, as a reference for searching other systems using these protocols. We summarise the different observed behaviour through a handful of points.

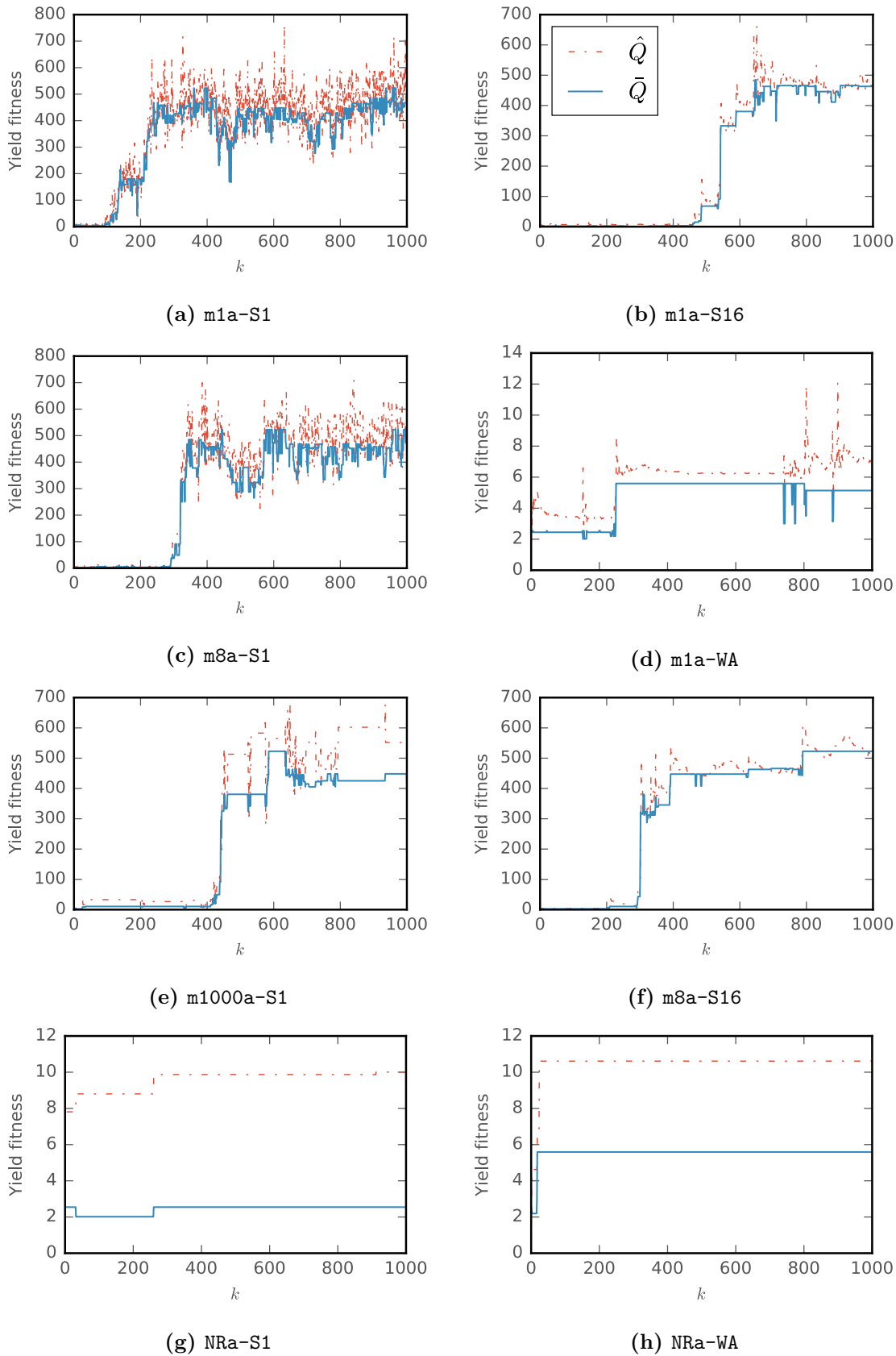


Figure 6.4: Fitness trajectories of example hill climbs of different protocols (see captions), with lines showing the true fitness $\langle \hat{Q} \rangle$ (blue solid) and measured fitness \hat{Q} (red dashed-dot) of the fitter candidate point at each iteration k . All start at $\mathbf{x}_0 = (10, 19)$ where the true fitness is 2.45.

6.4.1 Fitness changes in hill-climbing

The true and measured fitness changes, \bar{Q} and \hat{Q} , over the course of climbing trajectories for different protocols initiated at $(10, 19)$ and terminated after $k_{\max} = 1000$ are shown in Figure 6.4. Climbers were not necessarily close to each other at termination even if they had similar terminal fitness. In the plots on the left, showing climbs with no cumulative averaging (S1), \hat{Q} exhibited constant fluctuations at every iteration when $m = 1$ (m1) (Fig. 6.4a), but saw a slower rate of fluctuations for $m = 8$ (Fig 6.4c) as the number of trials dropped. By $m = 1000$ (Fig. 6.4e), fluctuations were sparse and changes in \bar{Q} also became less frequent. A climber was essentially stuck at a single state point for large intervals of k . Intervals of constant \hat{Q} were no greater than m iterations in length, but intervals of constant \bar{Q} exceeded m iterations when multiple trials at the same location are performed. The tendency to choose higher fitness in the move protocol resulted in $\hat{Q}(\mathbf{x}) > \bar{Q}(\mathbf{x})$ for most steps. Only **a** versions of the protocols are shown—there were no discernible differences found between individual trajectories of versions **a** and **b**.

Rapid jumps in yield fitness to levels of approximately 400 are indicative of a climber reaching the peak region, occurring at larger k as m is increased. Moreover, an increasing number of climbers failed to reach the global peak before termination. Climbers were found to be attracted to the secondary peak at $(7.2, 11)$ with $\bar{Q} = 270.2(1 \text{ d.p})$ when initialised near/at $(10, 19)$ (e.g. the fitness activity between $100 \leq k \leq 200$ in Fig. 6.4a). NR protocols (Fig. 6.4g and 6.4h) saw sparse changes in their monotonically increasing \hat{Q} trajectory, failing to exhibit any significant climbing activity—the majority of the climber population at termination was within two $\delta\epsilon$ and δw away from \mathbf{x}_0 .

Cumulative average protocols altered the fluctuations exhibited by \bar{Q} and \hat{Q} of the climbs. Comparing m1a-S1 (Fig. 6.4a) with m1a-S16 (Fig. 6.4b) and m1a-WA (Fig. 6.4d), a clear drop in the number of \hat{Q} fluctuations is seen. Relatively large \hat{Q} changes were observed when a climber evaluates an unvisited point \mathbf{x} . Subsequent evaluations of \mathbf{x} dampened the $\hat{Q}(\mathbf{x})$ fluctuations so longer intervals of constant \bar{Q} were observed. For WA, long intervals of being stationary in \mathbf{x} resulted in a converged \bar{Q} as $\text{history}(\mathbf{x})$ increased, resulting in even longer stationary intervals. Cumulative averaging had little effect on the frequency of yield fluctuations of NR protocols (Fig. 6.4h).

Overall, we see that the activity in fitness changes is a lot higher for low m and low r_{hist} values. Increasing m and r_{hist} values both independently reduce this activity, but does not necessarily cause a drastic degradation in terminal fitness performance when $\mathbf{x}_0 = (10, 19)$. The dependence of terminal fitness on \mathbf{x}_0 will be covered in the next section.

6.4.2 Averaged fitness performance for different protocols

Independent hill-climbing runs produce different outcomes. We capture the general effects of the different hill-climbing variants by considering the averaged \bar{Q} trajectory of the climber population. We expect \bar{Q} to increase as more climbers converge towards the global peak.

A set of 500 hill-climbing runs terminating after a maximum of $k_{\max} = 1000$ iterations were performed for each protocol combination described in Section 6.4 at various initial

points \mathbf{x}_0 . An overview of the average \bar{Q} outcomes for a selection of initial points \mathbf{x}_0 is found in Table 6.1, coarsely ordered by the expected memory used and level of book-keeping involved in the protocol definition (m1a-S1 does not track any yield values, NRa-WA holds all yield values throughout the climb). A subset of averaged yield trajectories are also plotted in Figure 6.5, chosen for their distinctive initial location on the yield landscape: (8.6, 4)—global maximum; (6, 10)—from fluid region via the secondary peak; (14, 2)—at a ridge; and (14, 19)—far away from the peak. Selected climber distribution plots showing terminal locations are shown in Figures 6.6 and 6.7 on pages 101 and 102.

Table 6.1: Average \bar{Q} at various iterations k for climbs starting at different initial points. The sets of hill-climbing protocols are categorised by the cumulative averaging scheme used (S1, S16, or WA). Each cell is coloured according to their normalised value within each column i.e. initial point. A darker fill indicates higher relative fitness.

| Protocol | | k | Initial point | | | | | | | | |
|----------|------|--------|---------------|--------|--------|---------|--------|---------|---------|--------|---------|
| | | | (6,2) | (6,10) | (6,19) | (8.6,4) | (10,2) | (10,10) | (10,19) | (14,2) | (14,10) |
| S1 | | | | | | | | | | | |
| m1a | 0 | 0.00 | 0.31 | 0.80 | 522.77 | 448.22 | 16.97 | 2.45 | 306.09 | 1.20 | 0.22 |
| | 50 | 49.27 | 141.93 | 55.30 | 440.93 | 401.47 | 353.17 | 10.81 | 311.30 | 224.03 | 1.00 |
| | 200 | 362.19 | 380.77 | 333.28 | 423.40 | 398.16 | 403.55 | 221.65 | 318.47 | 308.94 | 93.14 |
| | 1000 | 409.93 | 407.59 | 411.70 | 405.68 | 397.82 | 403.86 | 404.68 | 357.30 | 357.09 | 353.49 |
| m1b | 0 | 0.00 | 0.31 | 0.80 | 522.77 | 448.22 | 16.97 | 2.45 | 306.09 | 1.20 | 0.22 |
| | 50 | 65.17 | 173.16 | 74.19 | 444.74 | 406.43 | 370.06 | 13.67 | 317.88 | 268.27 | 1.32 |
| | 200 | 385.24 | 404.60 | 375.41 | 428.32 | 409.24 | 413.36 | 258.12 | 326.68 | 324.39 | 118.05 |
| | 1000 | 418.44 | 418.62 | 418.55 | 413.94 | 411.63 | 412.98 | 415.16 | 369.34 | 372.47 | 367.45 |
| m8a | 0 | 0.00 | 0.31 | 0.80 | 522.77 | 448.22 | 16.97 | 2.45 | 306.09 | 1.20 | 0.22 |
| | 50 | 45.88 | 134.86 | 48.77 | 446.85 | 407.76 | 354.68 | 7.73 | 316.07 | 234.60 | 0.71 |
| | 200 | 359.10 | 373.06 | 323.98 | 429.88 | 407.30 | 412.91 | 193.49 | 322.39 | 316.68 | 74.01 |
| | 1000 | 419.50 | 422.13 | 421.65 | 416.09 | 411.04 | 412.03 | 416.63 | 359.73 | 361.09 | 344.46 |
| m8b | 0 | 0.00 | 0.31 | 0.80 | 522.77 | 448.22 | 16.97 | 2.45 | 306.09 | 1.20 | 0.22 |
| | 50 | 59.48 | 156.43 | 61.45 | 453.09 | 411.15 | 368.94 | 8.26 | 323.29 | 271.17 | 0.77 |
| | 200 | 381.17 | 399.18 | 356.43 | 437.09 | 413.54 | 421.10 | 215.36 | 330.22 | 328.47 | 86.72 |
| | 1000 | 428.24 | 430.69 | 429.78 | 425.76 | 418.40 | 422.76 | 427.09 | 370.72 | 370.79 | 352.87 |
| m1000a | 0 | 0.00 | 0.31 | 0.80 | 522.77 | 448.22 | 16.97 | 2.45 | 306.09 | 1.20 | 0.22 |
| | 50 | 41.28 | 91.74 | 25.11 | 458.82 | 418.22 | 335.26 | 5.06 | 322.78 | 220.11 | 0.54 |
| | 200 | 285.48 | 261.43 | 160.84 | 443.13 | 415.93 | 401.46 | 76.83 | 327.87 | 310.93 | 22.47 |
| | 1000 | 389.90 | 395.02 | 358.32 | 433.68 | 420.66 | 419.53 | 304.94 | 345.76 | 344.38 | 170.39 |
| m1000b | 0 | 0.00 | 0.31 | 0.80 | 522.77 | 448.22 | 16.97 | 2.45 | 306.09 | 1.20 | 0.22 |
| | 50 | 50.51 | 108.66 | 32.11 | 460.51 | 420.87 | 348.05 | 5.81 | 327.38 | 252.47 | 0.68 |
| | 200 | 295.56 | 275.48 | 176.81 | 446.58 | 421.94 | 408.59 | 87.40 | 333.52 | 321.02 | 26.09 |
| | 1000 | 391.92 | 398.89 | 368.61 | 438.82 | 423.45 | 426.55 | 313.44 | 352.07 | 349.80 | 172.26 |
| NRa | 0 | 0.00 | 0.31 | 0.80 | 522.77 | 448.22 | 16.97 | 2.45 | 306.09 | 1.20 | 0.22 |
| | 50 | 36.92 | 51.50 | 17.06 | 457.47 | 420.54 | 298.42 | 3.13 | 320.97 | 175.60 | 0.36 |
| | 200 | 210.33 | 114.67 | 51.81 | 449.48 | 422.74 | 337.68 | 5.55 | 322.78 | 249.19 | 0.64 |
| | 1000 | 230.91 | 157.85 | 79.79 | 444.44 | 423.81 | 365.57 | 15.80 | 322.69 | 269.39 | 2.17 |
| NRb | 0 | 0.00 | 0.31 | 0.80 | 522.77 | 448.22 | 16.97 | 2.45 | 306.09 | 1.20 | 0.22 |
| | 50 | 44.59 | 60.02 | 21.28 | 457.99 | 421.41 | 308.35 | 3.15 | 321.23 | 201.23 | 0.36 |
| | 200 | 207.72 | 122.53 | 57.44 | 449.51 | 423.01 | 339.94 | 6.05 | 323.10 | 257.20 | 0.56 |
| | 1000 | 224.08 | 166.15 | 81.52 | 445.47 | 424.65 | 366.95 | 16.81 | 323.70 | 276.40 | 2.65 |
| S16 | | | | | | | | | | | |
| m1a | 0 | 0.00 | 0.31 | 0.80 | 522.77 | 448.22 | 16.97 | 2.45 | 306.09 | 1.20 | 0.22 |
| | 50 | 45.70 | 98.22 | 25.82 | 510.30 | 431.82 | 373.99 | 4.40 | 347.40 | 272.55 | 0.53 |
| | 200 | 239.81 | 253.01 | 148.27 | 513.93 | 448.61 | 445.44 | 98.43 | 360.90 | 354.24 | 23.83 |
| | 1000 | 334.30 | 371.15 | 357.13 | 516.26 | 486.86 | 489.83 | 364.22 | 376.41 | 376.94 | 196.98 |
| m1b | 0 | 0.00 | 0.31 | 0.80 | 522.77 | 448.22 | 16.97 | 2.45 | 306.09 | 1.20 | 0.22 |
| | 50 | 53.14 | 116.67 | 29.99 | 512.35 | 433.42 | 383.41 | 6.83 | 349.99 | 302.48 | 0.68 |
| | 200 | 243.16 | 265.71 | 156.39 | 515.08 | 452.04 | 449.56 | 120.01 | 362.85 | 357.50 | 29.18 |
| | 1000 | 322.00 | 367.39 | 354.48 | 517.48 | 490.25 | 492.36 | 371.02 | 376.60 | 377.37 | 192.78 |
| m8a | 0 | 0.00 | 0.31 | 0.80 | 522.77 | 448.22 | 16.97 | 2.45 | 306.09 | 1.20 | 0.22 |
| | 50 | 40.85 | 79.02 | 17.85 | 507.90 | 432.38 | 356.18 | 4.30 | 343.62 | 255.29 | 0.40 |
| | 200 | 234.81 | 201.03 | 92.11 | 512.93 | 442.71 | 429.41 | 47.50 | 357.36 | 347.54 | 9.25 |
| | 1000 | 301.44 | 327.71 | 291.72 | 515.97 | 474.75 | 478.97 | 302.05 | 372.23 | 372.03 | 134.31 |
| m8b | 0 | 0.00 | 0.31 | 0.80 | 522.77 | 448.22 | 16.97 | 2.45 | 306.09 | 1.20 | 0.22 |
| | 50 | 49.51 | 89.11 | 21.94 | 510.38 | 433.36 | 362.74 | 4.22 | 344.75 | 280.39 | 0.45 |
| | 200 | 237.27 | 200.51 | 97.63 | 514.07 | 442.95 | 429.88 | 46.64 | 358.88 | 351.76 | 9.41 |

Continued in next page.

| Protocol | k | Initial point | | | | | | | | | |
|----------|--------|---------------|--------|--------|---------|--------|---------|---------|--------|---------|---------|
| | | (6,2) | (6,10) | (6,19) | (8.6,4) | (10,2) | (10,10) | (10,19) | (14,2) | (14,10) | (14,19) |
| WA | 1000 | 297.85 | 330.58 | 294.57 | 516.85 | 475.22 | 477.63 | 304.31 | 372.43 | 372.99 | 135.78 |
| | m1000a | 0 | 0.00 | 0.31 | 0.80 | 522.77 | 448.22 | 16.97 | 2.45 | 306.09 | 1.20 |
| | | 50 | 39.33 | 44.04 | 11.39 | 497.68 | 429.93 | 317.81 | 3.20 | 329.20 | 212.98 |
| | | 200 | 200.97 | 75.71 | 24.18 | 499.76 | 432.53 | 341.96 | 5.47 | 333.03 | 276.65 |
| | | 1000 | 224.50 | 99.38 | 35.30 | 502.04 | 435.87 | 364.96 | 11.13 | 336.76 | 302.84 |
| | m1000b | 0 | 0.00 | 0.31 | 0.80 | 522.77 | 448.22 | 16.97 | 2.45 | 306.09 | 1.20 |
| | | 50 | 44.33 | 50.38 | 12.67 | 498.39 | 429.61 | 322.60 | 3.56 | 329.15 | 234.50 |
| | | 200 | 201.94 | 76.86 | 23.42 | 500.92 | 431.83 | 343.27 | 5.56 | 332.29 | 282.50 |
| | | 1000 | 223.81 | 100.28 | 32.84 | 502.63 | 435.47 | 364.15 | 9.08 | 336.71 | 305.09 |
| | NRa | 0 | 0.00 | 0.31 | 0.80 | 522.77 | 448.22 | 16.97 | 2.45 | 306.09 | 1.20 |
| | | 50 | 34.75 | 34.19 | 11.89 | 472.32 | 421.58 | 273.08 | 2.91 | 320.91 | 150.94 |
| | | 200 | 177.61 | 54.88 | 20.37 | 474.10 | 422.73 | 288.34 | 3.09 | 323.72 | 197.17 |
| | | 1000 | 199.43 | 73.34 | 28.10 | 475.81 | 424.47 | 302.86 | 3.78 | 326.34 | 221.89 |
| | NRb | 0 | 0.00 | 0.31 | 0.80 | 522.77 | 448.22 | 16.97 | 2.45 | 306.09 | 1.20 |
| | | 50 | 40.52 | 33.72 | 12.80 | 471.84 | 421.53 | 270.46 | 2.80 | 319.87 | 160.01 |
| | | 200 | 180.20 | 52.09 | 20.32 | 473.57 | 422.55 | 285.99 | 3.07 | 322.42 | 195.55 |
| | | 1000 | 202.67 | 72.27 | 27.41 | 475.41 | 424.36 | 301.93 | 3.23 | 325.32 | 220.51 |
| WA | m1a | 0 | 0.00 | 0.31 | 0.80 | 522.77 | 448.22 | 16.97 | 2.45 | 306.09 | 1.20 |
| | | 50 | 41.30 | 92.34 | 22.18 | 510.58 | 435.58 | 369.32 | 5.07 | 348.83 | 269.92 |
| | | 200 | 222.20 | 165.67 | 71.45 | 515.38 | 448.50 | 422.75 | 63.28 | 360.58 | 348.49 |
| | | 1000 | 241.22 | 185.23 | 87.86 | 516.88 | 451.51 | 430.87 | 157.04 | 362.95 | 356.01 |
| | m1b | 0 | 0.00 | 0.31 | 0.80 | 522.77 | 448.22 | 16.97 | 2.45 | 306.09 | 1.20 |
| | | 50 | 52.36 | 109.01 | 27.34 | 513.03 | 437.85 | 380.62 | 5.69 | 350.52 | 300.17 |
| | | 200 | 222.36 | 169.17 | 70.99 | 516.38 | 449.39 | 422.78 | 62.51 | 361.18 | 351.15 |
| | | 1000 | 237.88 | 185.91 | 85.81 | 517.19 | 452.04 | 430.72 | 149.54 | 363.29 | 357.44 |
| | m8a | 0 | 0.00 | 0.31 | 0.80 | 522.77 | 448.22 | 16.97 | 2.45 | 306.09 | 1.20 |
| | | 50 | 41.11 | 75.68 | 18.51 | 507.59 | 432.54 | 356.54 | 3.82 | 343.49 | 253.76 |
| | | 200 | 226.55 | 157.16 | 66.23 | 513.91 | 444.82 | 415.72 | 34.56 | 357.97 | 343.40 |
| | | 1000 | 249.06 | 181.50 | 88.29 | 516.67 | 451.53 | 430.16 | 130.19 | 362.93 | 356.47 |
| | m8b | 0 | 0.00 | 0.31 | 0.80 | 522.77 | 448.22 | 16.97 | 2.45 | 306.09 | 1.20 |
| | | 50 | 48.44 | 88.74 | 21.46 | 509.88 | 432.80 | 363.44 | 4.66 | 345.04 | 280.09 |
| | | 200 | 222.84 | 160.88 | 63.20 | 515.67 | 445.55 | 413.78 | 35.58 | 358.21 | 344.35 |
| | | 1000 | 242.19 | 184.85 | 85.84 | 517.72 | 452.26 | 426.93 | 117.37 | 363.00 | 355.70 |
| | m1000a | 0 | 0.00 | 0.31 | 0.80 | 522.77 | 448.22 | 16.97 | 2.45 | 306.09 | 1.20 |
| | | 50 | 39.69 | 45.11 | 11.29 | 497.46 | 429.79 | 320.42 | 3.15 | 329.50 | 216.44 |
| | | 200 | 196.85 | 64.78 | 20.53 | 498.78 | 430.75 | 333.22 | 4.61 | 330.93 | 265.04 |
| | | 1000 | 208.28 | 66.79 | 21.59 | 499.06 | 430.92 | 335.31 | 5.16 | 331.11 | 270.57 |
| | m1000b | 0 | 0.00 | 0.31 | 0.80 | 522.77 | 448.22 | 16.97 | 2.45 | 306.09 | 1.20 |
| | | 50 | 46.32 | 47.01 | 13.04 | 497.36 | 430.31 | 321.79 | 3.21 | 328.92 | 232.74 |
| | | 200 | 193.17 | 62.86 | 19.44 | 498.61 | 430.92 | 331.23 | 4.20 | 330.16 | 264.32 |
| | | 1000 | 203.76 | 66.05 | 20.58 | 498.88 | 431.10 | 333.06 | 4.67 | 330.38 | 268.04 |
| | NRa | 0 | 0.00 | 0.31 | 0.80 | 522.77 | 448.22 | 16.97 | 2.45 | 306.09 | 1.20 |
| | | 50 | 35.29 | 31.28 | 12.08 | 472.42 | 421.30 | 269.86 | 2.82 | 320.51 | 150.11 |
| | | 200 | 177.03 | 41.30 | 17.69 | 473.16 | 421.41 | 276.67 | 2.85 | 321.08 | 182.34 |
| | | 1000 | 186.74 | 43.31 | 18.36 | 473.30 | 421.45 | 277.63 | 2.92 | 321.15 | 186.16 |
| | NRb | 0 | 0.00 | 0.31 | 0.80 | 522.77 | 448.22 | 16.97 | 2.45 | 306.09 | 1.20 |
| | | 50 | 42.62 | 35.27 | 12.45 | 473.11 | 421.56 | 271.30 | 2.81 | 318.94 | 156.90 |
| | | 200 | 169.56 | 44.11 | 16.28 | 473.68 | 421.78 | 276.51 | 2.94 | 319.46 | 180.53 |
| | | 1000 | 179.33 | 45.62 | 17.31 | 473.91 | 421.83 | 277.41 | 2.94 | 319.53 | 183.69 |

Focusing on **S1** cases first, for most variants an exhaustive neighbourhood selection (**b**) led to better terminal yield than a neighbourhood selection with replacement (**a**). Protocols with low m have more explorative power, reaching high yield even when initialised (far) away from the global peak region (Figs. 6.6a, 6.7a, 6.7b). The skewness of the terminal climber distribution for low m reflects the underlying \bar{Q} landscape around the global peak region. Protocols with high m retained high \bar{Q} better at $k = 1000$ when initialised at a high \bar{Q} point already (Figs. 6.6d, 6.7c).

A reduction in the terminal \bar{Q} for all protocols initialised at high \bar{Q} points is seen for all m , as climbers moved away from the global maximum $\mathbf{x}^* = (8.6, 4)$ (e.g. Fig. 6.5a). Note the terminal yield for NR was lower at $k = 1$ due to a default initial yield of $\hat{Q}(\mathbf{x}_0) = 0$; NR climbers all move away from \mathbf{x}_0 for $k = 1$. The reduction in \bar{Q} by all protocols

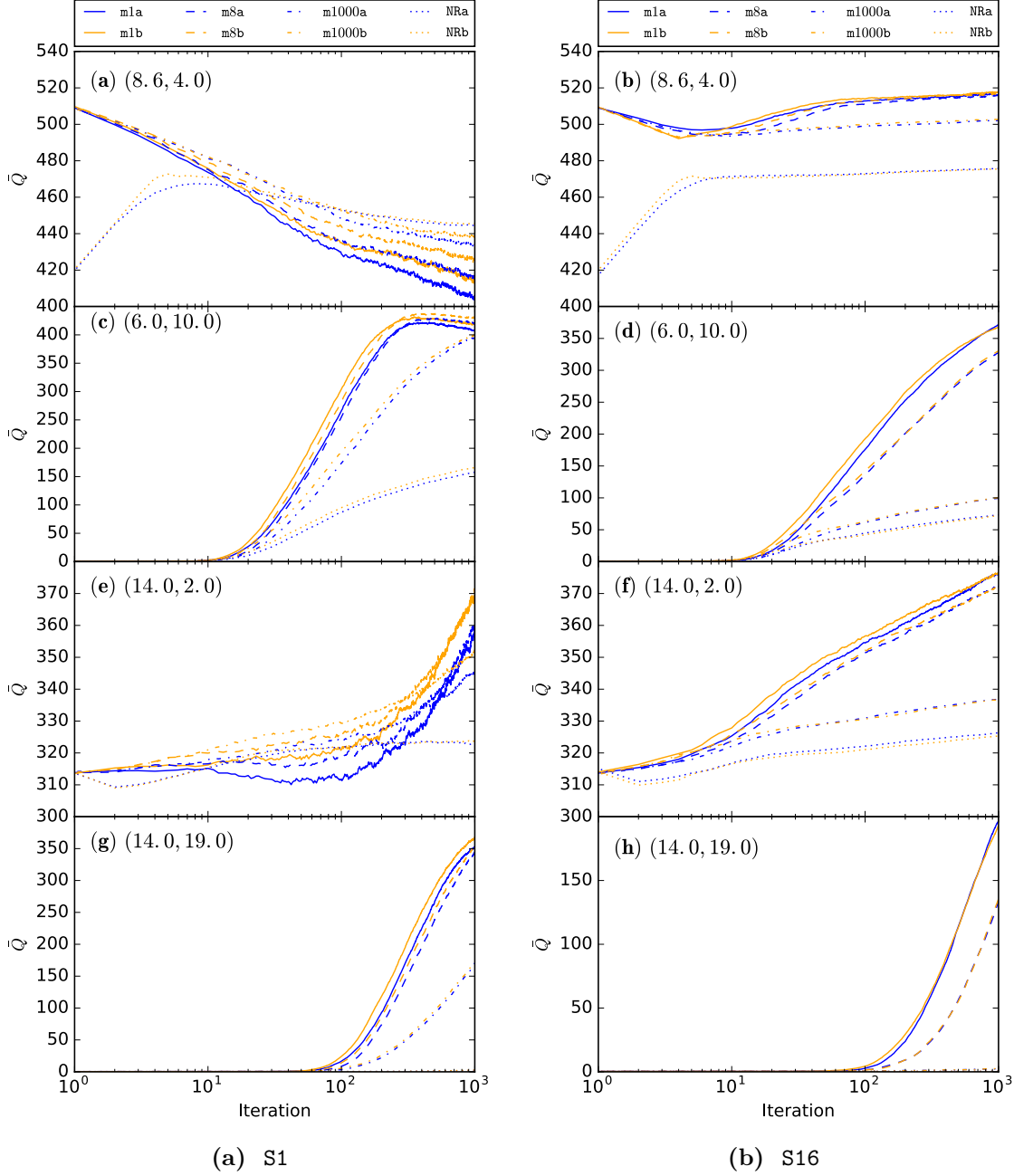


Figure 6.5: Plots of average true fitness \bar{Q} over 500 hill climbs at different initial points for (a) S1 and (b) S16 averaging schemes. Legend at top applies to all plots. Hill-climbing algorithms adopting protocol **b** (orange) consistent achieve higher fitness than the protocol **a** (blue) counterpart. For cumulative averaging with no memory limit (**WA**), the fitness changes are similar in form to $r_{\text{hist}} = 16$ case but with significantly lower increase for each case.

demonstrate the ‘melting of the peaks’—the effective annealing of peaks due to the noise enabling downhill moves—described by Levitan [144] (See Section 5.4.3). The $\mathbf{x}_0 = \mathbf{x}^*$ case gives us an idea of the best \bar{Q} attainable at termination, characterising the optimal steady state of the climber population on the \hat{Q} landscape. The average terminal fitness $\bar{Q}(\mathbf{x}_{k_{\max}})$ was higher for larger m (Fig. 6.5a), suggesting weaker annealing effects (Similar behaviour is found when comparing the distributions of Fig. 6.6a and Fig. 6.6d). Protocol NR resulted in the highest \bar{Q} .

Starting away from the global peak region, the terminal \bar{Q} was heavily influenced by the choice of m , but in general increased in fitness by the end of their runs. Infrequent retrialing still led to larger improvement in expected fitness when climbers were initiated at \mathbf{x}_0 only a small distance away from the hill and/or only encountered few local maxima to reach the global peak region (e.g. when $\mathbf{x}_0 = (10, 10)$ — See Fig. 6.6a and 6.6d). For $\mathbf{x}_0 = (14, 2)$ (Fig. 6.5c), m1 and m8 climber populations reached similar average terminal \bar{Q} to the $\mathbf{x}_0 = \mathbf{x}^*$ case (see Table 6.1). m1000 climbers had a lower average terminal \bar{Q} .

When \mathbf{x}_0 is far from the global peak region, climbers encountered more local maxima. m1a and m8a climbers yielded moderate average terminal \bar{Q} (Fig. 6.6c), whereas climbers adopting the m1000 or NR protocols resulted in suboptimal terminal \bar{Q} (Fig. 6.5e) or low terminal \bar{Q} (NR protocol in Fig. 6.5g).

Switching on the cumulative averaging (S16 and WA) either improved or worsened the terminal yield depending on the choice of \mathbf{x}_0 . Cases of \mathbf{x}_0 where the climber populations comfortably reached the global peak region in the S1 scheme saw an improvement in terminal fitness when cumulative averaging S16 was switched on (Figs. 6.5b, 6.5f, and Fig. 6.6b). $m = 1$ climbers stayed robust to the choices of \mathbf{x}_0 and cumulative averaging scheme. All other \mathbf{x}_0 points performed worse when S16 or WA schemes are adopted, with significantly less climbing activity seen for the respective m1000 and NR protocols. In these cases, climbers were concentrated in very few state points where local maxima lie, countering the annealing of the peaks, but often covered only a short distance from \mathbf{x}_0 (Fig. 6.6e, 6.7d, 6.7e, 6.7f). Interestingly, the cumulative averaging also affected the performance between a and b climbing performances. For WA, a climbers generally performed better than b. A possible reason for this is that the ability to pick the same neighbouring point in quick succession for a avoids the accumulation of too many samples in the time it takes to consider the same pair of state points, slowing down the rate of noise reduction.

We highlight the delicate balance in intensification and diversification provided by changing m and r_{hist} using Figs. 6.6a, 6.6b and 6.6c. When $\mathbf{x}_0 = (10, 10)$, m1a-S16 achieves this balance with most climbers reaching \mathbf{x}^* , compared to m1a-S1 that spreads out too much, and m1a-WA where answers are scattered in a few local maxima near \mathbf{x}^* .

In Table 6.1, the negative correlation between averaged \bar{Q} and m for WA protocols with a \mathbf{x}^* start is the reverse of the positive correlation seen for the S1a protocols. Three of the WA protocols are also displayed in Figs. 6.6g, 6.6h, and 6.6i, showing that less climbers are at \mathbf{x}^* when $m = 1000$ than when $m = 1$ or 8. We suspect that this negative correlation reflects how the cumulative averaging gradually freezes a climber’s activities. A climber with too strong a tendency to move uphill (e.g. $m = 1000$) saturates **history** of the neighbouring points faster as it gets stuck in these points, thus reducing the noise

(increasing the number of past samples in `history`) in the neighbouring points faster than the climbers are able to return to \mathbf{x}^* . Again, this highlights the delicate interaction between these two metaheuristics with the landscape. Figs. 6.7d and 6.7e illustrate how the prominent secondary peak hinders climbers initiated around the low ε high w region from converging to the global peak region.

To better understand how the averaged terminal \bar{Q} depends on m , we study the \bar{Q} distribution of the climber populations in the next section.

6.4.3 Influence of m on terminal fitness distribution of climbers

In the previous section, we saw that the averaged terminal fitness was dependent on m . To look further into why this is the case, we study the distribution of climber fitnesses.

The late-time \bar{Q} distributions for climber populations with `S1` and `a` but different m values are plotted in Figure 6.8 on page 103. Focusing on the climbs initialised at the global peak \mathbf{x}^* (blue circles) only, the late-time \bar{Q} distribution was skewed towards high \bar{Q} values when $m = 1$, with most climbers in the bin centred at $\bar{Q} = 460$. This distribution becomes increasingly skewed as m is increased due to more climbers having yield values in the range 430 – 470 (Also refer to Fig. 6.6a and 6.6d, showing a similar terminal climber distribution). These distributions approximate the best terminal climber distribution for each m if all climbers reached the global peak region.

Next, we compare these distributions with those initialised at $\mathbf{x}_0 = (6, 19), (14, 19)$. Climbers with $m = 1$ and $m = 8$ performed well when $\mathbf{x}_0 = (6, 19)$, converging to the same distribution, but could not cope when $\mathbf{x}_0 = (14, 19)$, with a large spread in \bar{Q} values. Some climbers still had $\bar{Q} < 10$. The late-time \bar{Q} distribution was subject to \mathbf{x}_0 when $m = 1000$. Around 10% of climbers have $\bar{Q} > 10$ when $\mathbf{x}_0 = (6, 19)$, rising to around half the population when $\mathbf{x}_0 = (14, 19)$. The remaining climbers have fitnesses in localised pockets of yield values where they are stuck in local maxima. For $\mathbf{x}_0 = (6, 19)$, climbers stuck in the secondary peak (see Fig.4.8) contribute to the spike in the $\bar{Q} = 280$ bin.

6.4.4 Discussion

The runs on \hat{Q} so far suggest that hill-climbing can find the global peak region of a yield landscape, and thus the optimal or near optimal interaction parameters for the targeted assembly of a self-assembly system. The spread of the terminal climber locations highlight the annealing of the peaks, suggesting that hill-climbing is more suited for finding the global peak region rather than the global peak. While protocols `a` and `b` produced little difference in the climbing behaviour, both trialing the current fitness and cumulative averaging led to different results. Cumulative averaging self-evidently provided intensification to a hill climb which induced a poor climbing behaviour when climbers were initialised far away from the peak, allowing a climber to localise its exploration to a smaller region near a local maximum. Surprisingly, the trialing of current fitness values also exhibited intensification effects, although much weaker, as seen in the gathering of climbers in $m = 1000$ in Fig. 6.8 and when comparing the terminal climber distributions in Figs. 6.6a and 6.6d. We will understand why this is the case in the next section.

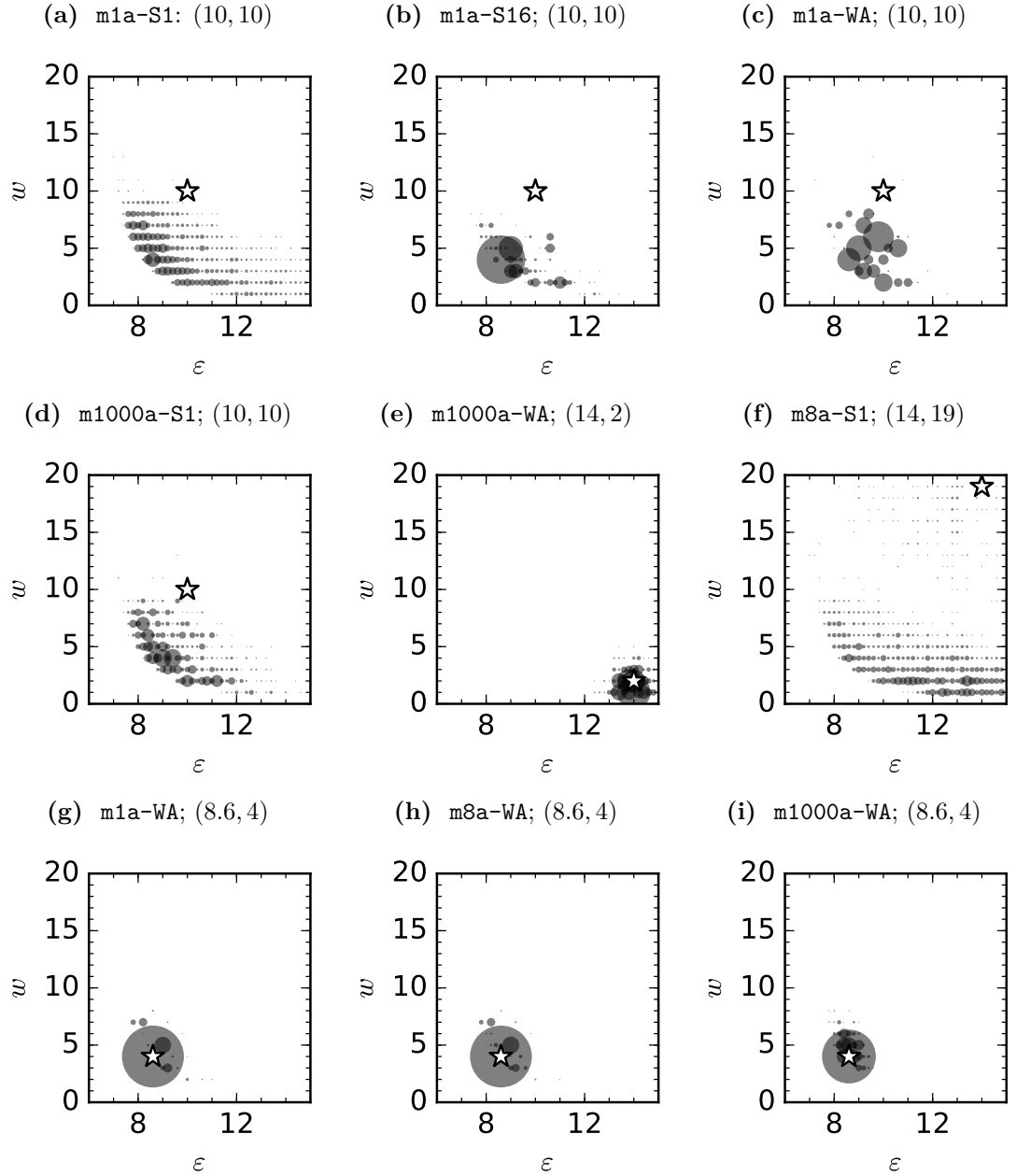


Figure 6.6: Plots of terminal climber distribution on \mathcal{P} . Circle markers show where climbers are found, semi-transparent to show overlapping markers. Marker area scales linearly with number of climbers found at each point. Corresponding hill-climbing protocol and initial point \mathbf{x}_0 (star) are given in caption. (a),(b),(c) demonstrate the necessary use of averaging to achieve smaller spread in climber distribution, but can result in too much trapping at local maxima as r_{hist} is increased. (a) and (c) highlights the weak intensification effects introduced when m is increased. (e) shows poor climbing ability when m and r_{hist} are both too high. (f) illustrates the high explorative power required by climbers to traverse a plateau region. (g),(h),(i) highlights the sensitive balance between r_{hist} and m ; see main text.

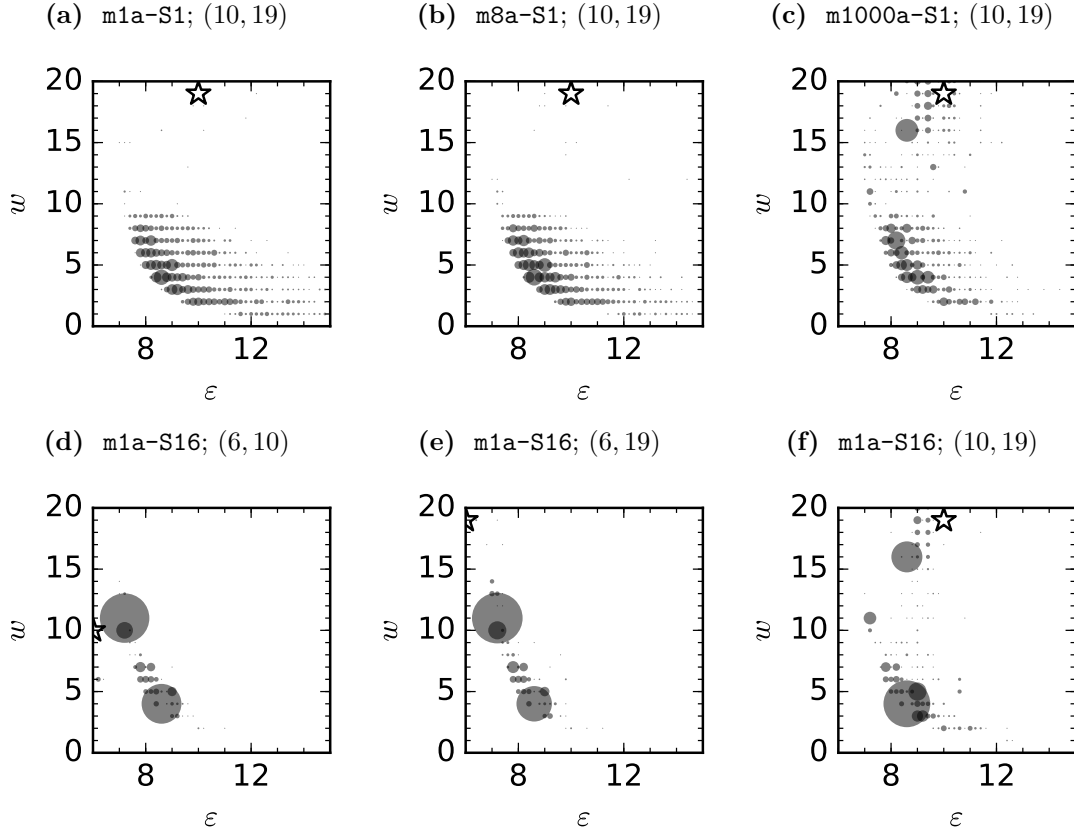


Figure 6.7: Plots of terminal climber distribution on \mathcal{P} . Circle markers show where climbers are found, semi-transparent to show overlapping markers. Marker area scales linearly with number of climbers found at each point. Corresponding hill-climbing protocol and initial point \mathbf{x}_0 (star) are given in caption. Marker area scales linearly with number of climbers found at point. (a),(b),(c),(f) illustrate how high m and high r_{hist} both independently increase intensification but decrease diversification, reducing search robustness. (d),(e) shows the secondary peak hindering convergence, and is an example of a major obstacle shared by close-lying \mathbf{x}_0 to yield similar expected terminal behaviour. (d),(e),(f) show how climbers may gather at drastically different \mathbf{x} , compromising reliability and robustness.

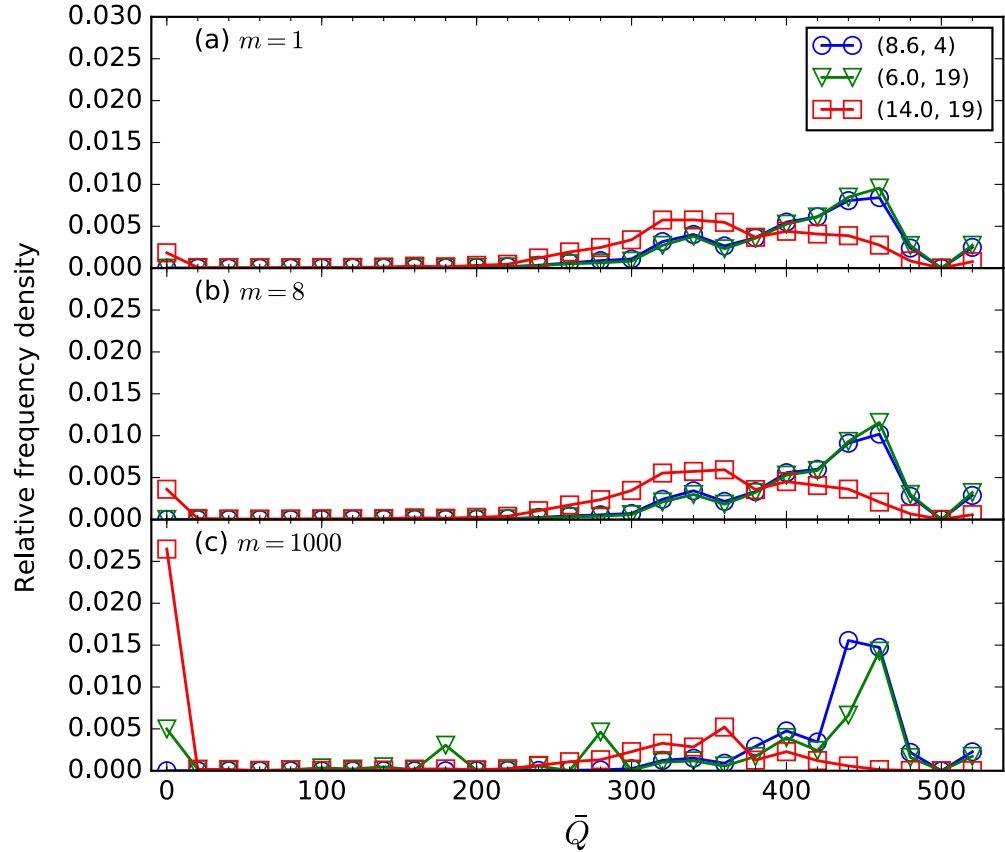


Figure 6.8: Histogram of the true fitnesses of multiple sets of S1a climbs between $800 \leq k \leq 1000$ for (a) $m = 1$, (b) $m = 8$, and $m = 1000$, each set starting at either $(6, 19)$ (*triangle*), $(14, 19)$ (*square*), or the global maximum $\mathbf{x}^* = (8.6, 4)$ (*circle*). Bins are centred at multiples of 20, with bin widths of 20. For each population of 4000 climbers, \bar{Q} at step k is tallied to produce a histogram. This is performed separately at $k = 800, 900, 1000$ to then obtain a bootstrapped mean of them at each bin. Error bars are smaller than the marker size for all points. \bar{Q} at \mathbf{x}^* is 522.7 (1 d.p.).

Effective \mathbf{x}_0 choices within \mathcal{P} led to efficient and reliable search outcomes for all hill-climbing variants. A random initialisation of the climb calls for a low m and low r_{hist} to ensure consistent convergence to the global peak region. The original stochastic hill-climbing algorithm (equivalent to **m1a-S1**) is thus the most robust amongst the variants studied due to its explorative power, is reliable in reaching the global peak region but has relatively low accuracy in terminal result as it does not converge to a localised region once at the global peak region. Compare this to an increased m or r_{hist} protocol, where hill-climbing becomes less robust (climbing performance is reliant on choice of \mathbf{x}_0), less reliable (as climbers may end up in drastically different parts of \mathcal{P} e.g. Fig. 6.7c), but more accurate (in the sense that the climbers are localised within the respective regions, and thus consistency in the result is guaranteed *if* the correct basin of attraction is found). The balance between intensification and diversification is controlled by the balance in r and r_{hist} . It is therefore worth considering an scheduled implementation of the current fitness trial period and cumulative averaging, where increasing m and r_{hist} values gradually provides the best opportunity for a climber to both reach the global peak region robustly and improve the yield value of the solution at termination. This would achieve a similar effect to increasing the evaluation sample size r , but without increasing the computational intensity of the search (in fact the computational intensity would be decreased).

The least accurate \hat{Q} approximations of the noise fluctuations in a simulated Q were found for low \hat{Q} values. Certain state points were therefore consistently local maxima, which led to some climbing trajectories and behaviour being favoured locally, especially at low \hat{Q} value. We do not expect this to have a significant effect on the qualitative interpretation of the trajectories and its version executed with VMMC simulation, particularly at high yield values where accurate approximations are found.

The standard stochastic hill-climbing algorithm (**m1a-S1**) is able to consistently reach the global peak region within 300 iterations, with the final search outcome robust to the choice of \mathbf{x}_0 . On the other hand, increasing m or r_{hist} reduces the robustness of the algorithm as climbers became less proficient at finding the global peak when initiated far away from the global peak region. In terms of reliability, measuring the variability of the outcome, a balance in the protocols seems best. **m1a**, although often reaching the same distribution, has a broad spread in terminal fitness and location within the local region. **m1000a** has more confined spreads, but is distributed across the parameter space. Compare this to the densely populated points achieved when increasing r_{hist} , which enables over 95% of the climber population to converge to a single point at termination in some cases. This highlights the necessity for fitness averaging over multiple samples, through the adoption of cumulative averaging or increased sample size per evaluation, as the only way to gain more accurate and reliable search outcomes.

6.5 Move probabilities

In Section 5.4.3 we covered how Levitan’s study [144] of the NK model showed that a decreased probability of finding fitter neighbours along with a decreased average difference in fitness between neighbours as the fitness value becomes higher both contribute to the

initial fast rise and gradual plateauing of the fitness trajectory of an adaptive walk in the noiseless version of the model. Meanwhile, additional probabilities were considered for a noisy NK model landscape, characterising the likelihood of an accepted move coinciding with the same move type (uphill or downhill) on the underlying fitness landscape. Although this explains why increasing the noise variance reduces the average fitness reached by the walkers, these probabilities fail to capture the behaviour of reusing fitness measurements in a trial and why increasing the trial period limit m reduces the annealing effects of noisy local search. In this section, rather than focusing on fitness-dependent move probabilities, we consider fluctuation-dependent move probabilities to elucidate the influence m has on the hill-climbing behaviour.

Recall that in a noisy landscape, the measured fitnesses $g(x)$ and $g(x')$ of two neighbouring points x and x' are subject to two factors: the difference Δf between the underlying true fitness values $f(x)$ and $f(x')$; and the noise contributions $Y(x)$ and $Y(x')$. We consider the move probabilities *after* a climber has evaluated its current position x to yield a measured current fitness $g(x) = f(x) + Y(x)$, i.e. the probabilities given $Y(x)$.

According to Algorithm 3 on page 92, the condition for a *successful move* is

$$\Delta g = \Delta f + Y(x') - Y(x) \geq 0, \quad (6.3)$$

which gives

$$Y(x') \geq Y(x) - \Delta f. \quad (6.4)$$

$\Delta g < 0$ results in a *failed* or *unsuccessful* move attempt. Recall that a successful move can be uphill or downhill in f [144]. We call a successful move that is also an uphill move in f a *correct* move ($\Delta f > 0$), and one that leads to a decrease in f an *incorrect* move ($\Delta f < 0$)—this is equivalent to the true positive (*true+*) and false positive (*false+*) moves mentioned in Section 5.4.3. We consider neutral moves in true fitness ($\Delta f = 0$) as correct moves as well.

For a neighbour $x'_j \in \mathcal{N}(x)$, let the probability of a move occurring given $Y(x)$ be

$$\begin{aligned} P_j(Y(x)) &= P(Y(x'_j) > Y(x) - \Delta f_j) \\ &= \int_{Y(x) - \Delta f_j}^{\infty} P(Y(x'_j)) dY(x'_j), \end{aligned} \quad (6.5)$$

where $P(Y)$ is the probability density for sampling g with fluctuation Y .

The probability of a successful move given $Y(x)$, for a fitness value f , is

$$P(\Delta g \geq 0 | Y(x), f) = \frac{\sum_j P_j(Y(x))}{\sum_j \int_{-\infty}^{\infty} P(Y(x'_j)) dY(x'_j)}, \quad (6.6)$$

where the sums are over the available neighbouring points $x'_j \in \mathcal{N}(x)$. The domain of $P_j(Y(x))$ decreases as $Y(x)$ increases, thus $P(\Delta g \geq 0 | Y(x), f)$ drops when $Y(x)$ is higher. Intuitively, it is likelier to have $g(x') \geq g(x)$ if $Y(x)$ is more negative (See Fig. 6.9). The form of this probability curve is dependent on the average Δf between neighbouring points found at points with fitness value f in the same manner as in Ref [144]. However,

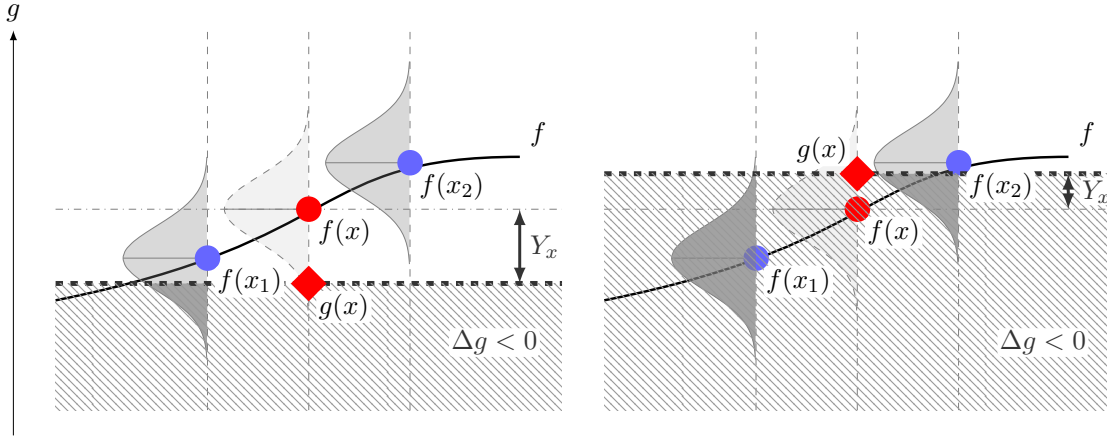


Figure 6.9: Schematic illustrating the likelihood of outcomes for a climber moving away from a point x after its current fitness $g(x) = f(x) + Y_x$ is evaluated. The point x has two neighbouring points x_1 and x_2 with fitnesses $f(x_1)$ and $f(x_2)$ respectively and noisy evaluations g represented by the normal distribution curve at the points on the fitness curve f . A larger proportion of move attempts, where a random neighbour is chosen and its fitness $g(x')$ evaluated, will be a successful move ($\Delta g > 0$) for negative Y_x (left diagram) than for positive Y_x (right diagram). On the other hand, a successful move is likelier to have led to a correct move ($\Delta g > 0$ and $\Delta f > 0$) than an incorrect one ($\Delta g > 0$ but $\Delta f < 0$) if Y_x is more positive than negative, illustrated by the relatively larger proportion of noisy distribution above $g(x)$ between points x_1 and x_2 when Y_x is positive.

we expect to observe the same behaviour for each f value.

The probability $P(\Delta f \geq 0 | \Delta g \geq 0, Y(x), f)$ that, given fitness f , a successful move is a correct one is also dependent on the fluctuation $Y(x)$, given by

$$\begin{aligned}
 P(\Delta f \geq 0 | \Delta g \geq 0, Y(x), f) &= \frac{\sum_{j, \Delta f_j \geq 0} P_j(Y(x))}{\sum_j P_j(Y(x))} \\
 &= \frac{\sum_{j, \Delta f_j \geq 0} P_j(Y(x))}{\sum_{j, \Delta f_j \geq 0} P_j(Y(x)) + \sum_{j, \Delta f_j < 0} P_j(Y(x))}, \quad (6.7)
 \end{aligned}$$

where the summation subscript $\Delta f_j \geq 0$ ($\Delta f_j < 0$) indicates the summation of neighbours with higher (lower) fitness than x . This is the proportion of the probabilities $P_j(Y(x))$ between $\Delta f_j > 0$ and $\Delta f_j < 0$ neighbours (See Fig. 6.9). Note that integrating Eqn. (6.7) over all Y for a given fitness value yields $P(\Delta f > 0 | \Delta g > 0, f)$, which is what Levitan studied (except the handling of neutral moves).

Equation (6.6) and (6.7) was sampled over the course of several hill climbs on \hat{Q} , using the normalised fluctuation $Y^* = Y(\mathbf{x})/\sigma(\mathbf{x})$ since the fluctuations vary with \mathbf{x} , effectively integrating Equation (6.6) over a selected range of fitness values f , to yield the plots in Figure 6.10. In Figure 6.10a, the probability of successful moves is almost one for highly negative Y^* , dropping to zero as expected when Y^* is highly positive. Meanwhile, only approximately 40% of successful moves are correct when Y^* is highly negative, but this increases up to 1 for highly positive Y^* (Fig. 6.10b). Large variations at the tails of these curves are due to the small sample size and the different truncations of the TND

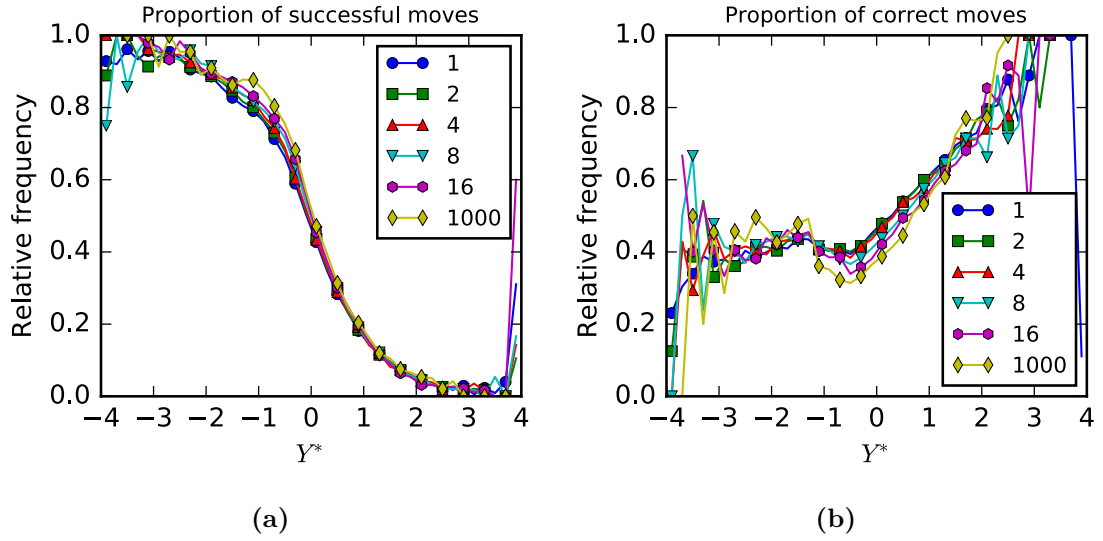


Figure 6.10: (a) Plot of $P(\Delta g > 0, Y^*(\mathbf{x}))$ for a typical hill climb. Legend corresponds to m . (b) Plot of $P(\Delta f > 0 | \Delta g > 0, Y(\mathbf{x}))$. Legend corresponds to m .

distributions at each point.

Note that the overall probability of seeing correct moves for a fitness value f is given by

$$P(\Delta f \geq 0, f) \propto \int P(\Delta f \geq 0 | \Delta g \geq 0, Y, f) P(\Delta g \geq 0 | Y, f) P(Y) dY. \quad (6.8)$$

Crucially, these probabilities are independent of m , in agreement with the lack of effect of m observed in Figs. 6.10a and 6.10b.

To consider the effects of m , the probability of a trial resulting in a successful move, a successful trial, is examined:

$$P(\text{successful trial} | Y(x), m) = 1 - (1 - p)^m. \quad (6.9)$$

At the cost of more move attempts (iterations) per sampled Y , moves with more positive Y have a relatively higher improved chance of being successful when m is increased.

Figure 6.11a shows a histogram of the relative frequency of Y^* per iteration. The distribution for $m = 1$ reflects the averaged TND noise distribution of the evaluations, peaking around $Y^* = 0$. Increasing m effectively shifted this distribution in the positive direction as higher Y^* was reused more often, such that moves with a higher chance of being a correct one were attempted more frequently relative to its occurrence in evaluations. In Figure 6.11b, the relative frequency of successful moves peaks at $Y^* = -0.5 \pm 0.2$ for $m = 1$. Increasing m again shifted this peak to the right, but also scaled down the total number of successful moves. Consequently, fewer downhill moves were performed as m was increased.

6.5.1 Discussion

This section showed that the dependence of the hill-climbing algorithm on the fitness function shown by Levitan [144] is still the crucial factor in the behaviour of modified

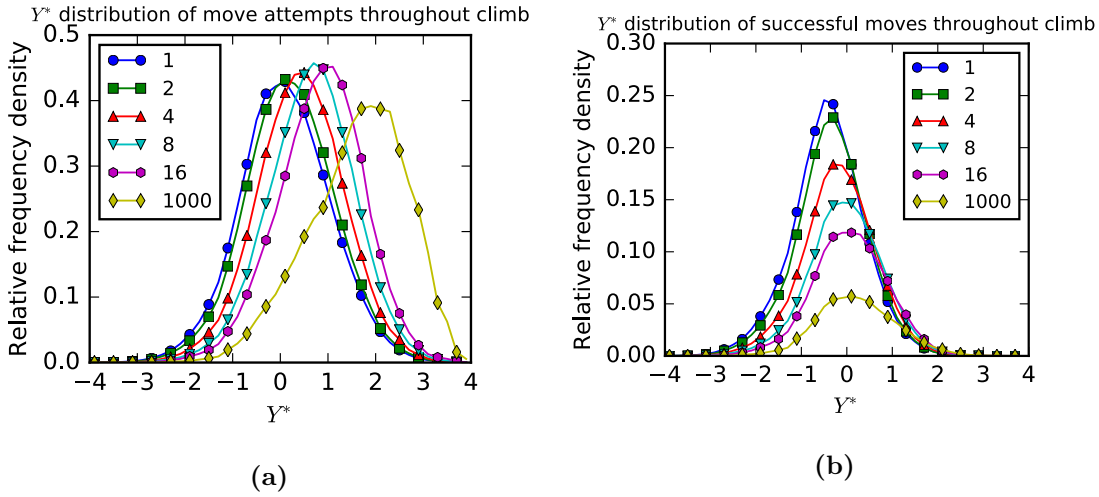


Figure 6.11: (a) Plot of relative frequency of Y^* appearing in each iteration in a typical climbs. Legend corresponds to m values. (b) Plot of relative frequency of successful moves, $P(\Delta g > 0, Y(\mathbf{x}))$, in a typical climb. The area under each curve represents the total number of successful moves, decreasing as m increases. The peak shifts away from negative $Y(\mathbf{x})$, where accepted moves are likelier to be downhill moves. Legend corresponds to m values.

hill-climbing algorithm. Larger f values have reduced $P(\Delta f > 0 | \Delta g > 0, f)$ as it becomes likelier for climbers to move downhill due to noise. The maximum trial period limit m merely enhances these chances relatively, suppressing the likelihood of moving away from the local maxima, but cannot overcome small initial probabilities.

Almost all attempts are accepted if the current evaluation $g(x)$ has a highly negative fluctuation Y^* , whereas failed moves are likelier if the relative fluctuation Y^* is large and positive. A move attempt with more positive Y^* has a lower chance of being successful, but is likelier to result in climbing uphill in f should a move occur. When m is too large it has a negative effect on the climbing behaviour as the drop in successful moves is not worth the evaluation cost. Striking a balance in the choice of m would combine the best of searchability and efficient exploring of the space. The trial period limit m works differently to the standard approach of reducing noise variance by increasing the evaluation sample size r , providing an independent way to manipulate the noise seen by a climber.

6.6 Climbing to threshold fitness

Having looked at how m affects the hill-climbing protocol, we understand that increasing m reduces the number of downhill moves at the cost of having to fail more move attempts. The sweet spot for m is found when a sufficient increase in the number of correct moves is achieved without expending too many attempts deemed likely to fail. More specifically, there is an optimal choice of trial period limit m^* where on average, after m^* consecutive failed move attempts using the same measured current fitness $\hat{Q}(\mathbf{x})$, it is more beneficial to reevaluate the current fitness. In this section, we empirically find m^* for the modified stochastic hill-climbing algorithm with protocols a and S1 on the yield landscape model

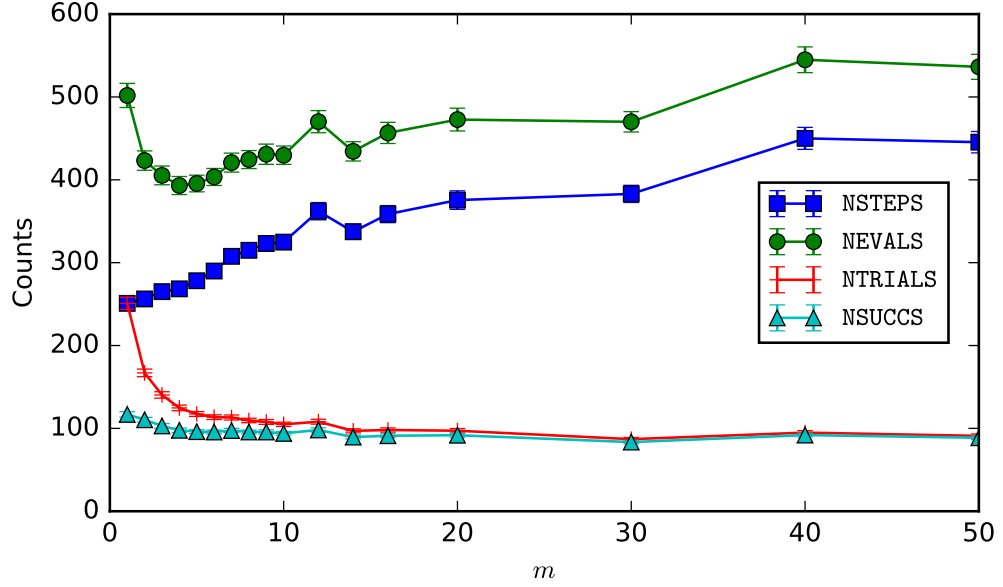


Figure 6.12: Plot of various counts for threshold climbs. For each m value, all 500 climbers start from the same point, and are terminated upon reaching a fitness $\langle \hat{Q} \rangle \geq \bar{Q}^* = 370$. The number of iterations (NSTEPS); evaluations (NEVALS); trials (NTRIALS); and successful moves (NSUCCS) required for climbers to reach a value of \bar{Q}^* are considered and averaged.

\hat{Q} by performing climbs from a low initial yield value and observing how quickly a chosen fitness threshold \bar{Q}^* is reached. This is especially relevant for searches where a high self-assembly yielding region of parameter space is desired rather than the maximum, thus locating the global peak region suffices, or when it is used as part of a system of search techniques (e.g. hill-climbing before a high-resolution brute-force scan). Again, this is complementary to the idea of searching under practical constraints.

6.6.1 Experimental details and results

To perform a hill-climbing run to threshold fitness \bar{Q}^* , a climber is initialised at point $\mathbf{x}_0 \in \mathcal{P}$ with $\bar{Q}(\mathbf{x}_0) < \bar{Q}^*$, and allowed to climb \hat{Q} until it reaches a true fitness value $\bar{Q} \geq \bar{Q}^*$ for the very first time, at which point the search is terminated. Each experiment in this section were performed for a set of 500 independent threshold climbing runs, before taking an average over each set of observations.

Figure 6.12 shows the averaged observations for a set of threshold runs initialised from $\mathbf{x}_0 = (14, 19)$, where $\bar{Q}(\mathbf{x}_0) \approx 0.22$, and terminated when a climber reached $\bar{Q}^* = 370$, or terminated prematurely when k exceeded $k_{\max} = 10^5$ iterations. Only 6.5% of points in \mathcal{P} have $\bar{Q} \geq 370$. The average number of steps taken to reach \bar{Q}^* , NSTEPS, increased monotonically as m increased, indicating that longer hill climbs were required to reach the same fitness threshold. Contrast this with the asymptotically decreasing NTRIALS, the number of trials performed during the threshold run. For $m = 1$, NTRIALS=NSTEPS since a climber reevaluates its current fitness regardless of the outcome of the preceding move. As m increased, climbers performed fewer trials to reach \bar{Q}^* , exhibiting a rapid initial decrease

(the number of trials is almost halved when $m = 3$) before converging asymptotically to NSUCCS (the number of successful moves executed) since a climber is forced to evaluate \hat{Q} after a successful move. NSUCCS also indicates the distance travelled by a climber. The decrease in NSUCCS as m was increased suggests a more direct convergence to \bar{Q}^* with fewer downhill moves.

The total number of evaluations performed is given by NEVALS ($\text{NEVALS} = \text{NSTEPS} + \text{NTRIALS}$). From $\text{NEVALS}=620$ at $m = 1$, NEVALS reduced initially to its lowest value of 535 at $m = 3$ (roughly 80% drop), before steadily rising as m continued to increase. By $m = 40$ more evaluations were required to reach the threshold fitness than when $m = 1$. For convenience we will refer to the decreasing part of the NEVALS curve at small m as the *dip* and the increasing part after the minimum value as the *tail*.

Threshold hill climbs were also executed for different initial points \mathbf{x}_0 and evaluation sample sizes r , and for an additional threshold fitness value of $\bar{Q}^* = 250$. 19.5% of points in \mathcal{P} have $\bar{Q} \geq 250$. NEVALS differed in magnitude for different combinations of \mathbf{x}_0 and r , an amalgamation of starting at various distances from high fitness and having different climbing behaviour. To isolate the effect of m on hill-climbing performances under all these different circumstances, the number of evaluations is normalised by the $m = 1$ case for each experiment to observe the relative changes in NEVALS, shown in Figure 6.13 on the facing page.

Similar to Figure 6.12, non-monotonic increases in NEVALS were observed for all conditions tested, and consistent across the two chosen \bar{Q}^* . Focusing on $\bar{Q}^* = 370$, in Figure 6.13a where $r = 1$, the form of the NEVALS curve was similar to that found in Figure 6.12 (forms in NSTEPS and NTRIALS curves were also similar (not shown)). A minimum in NEVALS of up to 76% was found in NEVALS for integer $m \in [2, 10]$. The drop in NEVALS existed for all tested m values between 2 and 50 compared to the $m = 1$ case when initialised at (6, 2), but was only found to improve NEVALS for a narrow window of $2 \leq m \leq 30$ when initialised at (10, 19), and at an even narrower window $2 \leq m \leq 10$ when initialised even further away at (14, 19). The dip lasted for a shorter number of m , while the tail of NEVALS was much steeper when \mathbf{x}_0 was further away from high fitness.

In Figure 6.13b, climbers were all initialised from (10, 14). For this set of threshold runs the sample size of each evaluation was varied to test the effect of m on climbs for different levels of noise fluctuations (increasing r reduces σ). Each NEVALS curves exhibited a relatively steeper increasing tail and shorter dip as r increased.

Surveying the other attributes, the increase in NSTEPS is responsible for the form of the NEVALS tails in all cases. NTRIALS consistently converged asymptotically to approximately 40% of the $m = 1$ case across all conditions except for $r = 8$, where there was a convergence down to 20%. Overall, for different \mathbf{x}_0 and r tested, and for threshold fitnesses $\bar{Q}^* = 250$ and 350, $m^* = 3$ gave the most consistent reduction in NEVALS, being at $80\% \pm 3\%$ of the $m = 1$ case.

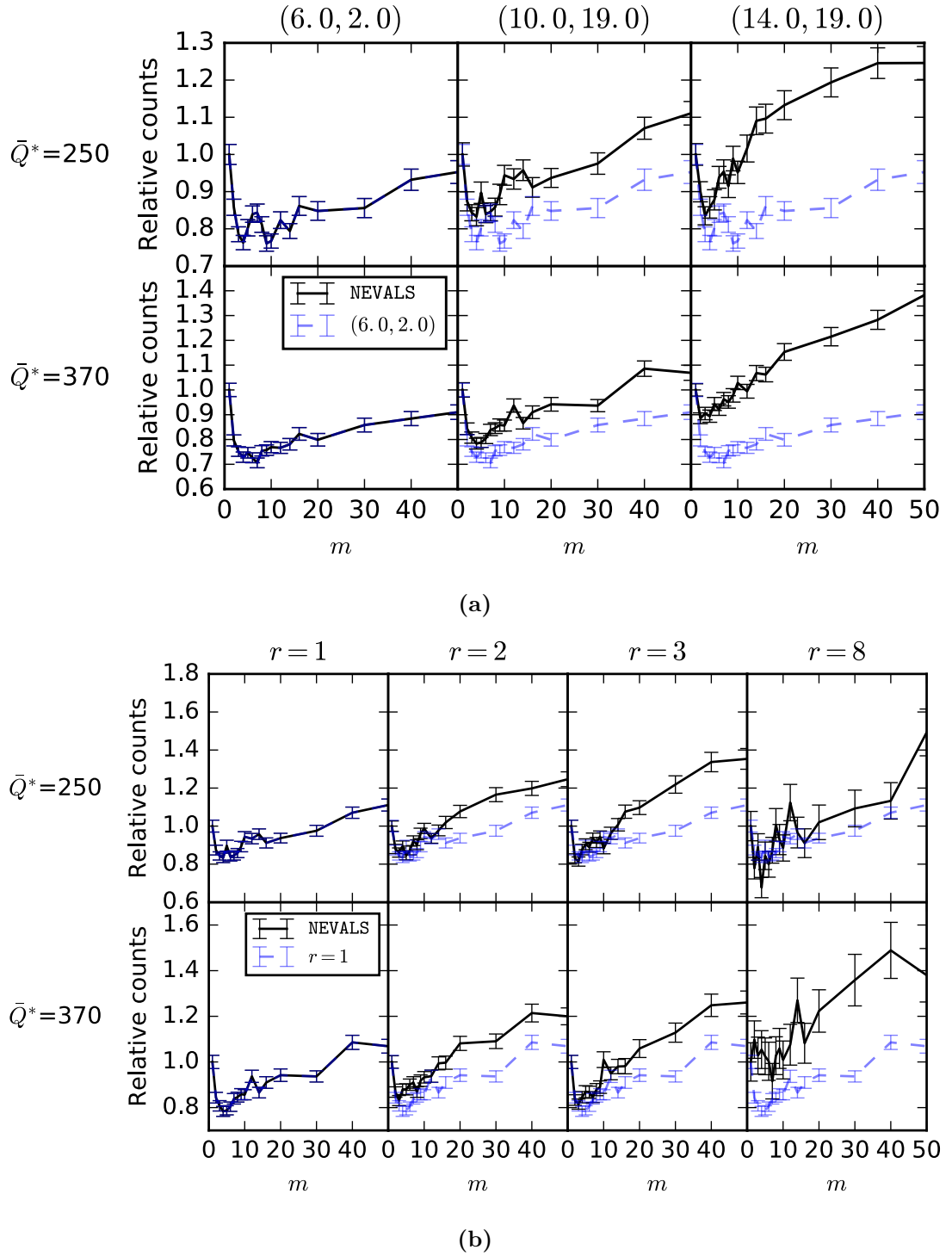


Figure 6.13: Plots of relative number of evaluations NEVALS taken to reach threshold fitness \bar{Q}^* for a range of maximum trial periods m between and including 1 and 50 for (a) different initial points (columns), and (b) for different evaluation sample size r (columns). NEVALS for different m cases are normalised by the $m = 1$ case. 500 climbs were performed for each set of runs. In (a), the initial points $(\varepsilon, w) = (6, 2)$, $(10, 19)$ and $(14, 19)$ get further away from the global peak \mathbf{x}^* , but all have $\bar{Q} < 2.5$. In (b), climbs were all initialised at $(10, 19)$. The solid line (black) is a guide, and the dashed line (blue) is a copy of the left-most case on the same row, redrawn for comparison.

6.6.2 Discussion

NEVALS signifies the computational effort required for a climber to reach threshold fitness, while NSTEPS indicates the minimum execution time required. An overall drop in NEVALS is possible when the rapid decrease in NTRIALS outweighs the small increase in NSTEPS at low m . However, a climber with an m value too high wastes iterations. The effectiveness of reusing the same $\hat{Q}(\mathbf{x})$ thus drops, requiring more iterations to reach \bar{Q}^* . This is the effect of what we observed in the previous section, where increasing m improves the outcome of successful moves, but at the cost of failing more attempts overall.

For users looking to select m different from unity, a consideration of a potential reduction in the computational effort t_{TOTCPU} must be made in conjunction with the inevitable increase in hill-climbing length t_{EXEC} . Since the hill-climbing algorithm is an iterative improvement algorithm, each iteration must be completed before moving onto the next. If two evaluations of an iteration can be performed concurrently, then NSTEPS dictates the minimum execution time $t_{EXEC} = \sum_k \max\{t_{EXEC,g(x_k)}, t_{EXEC,g(x'_k)}\}$ of the search (where $t_{EXEC,g(x_k)}$ and $t_{EXEC,g(x'_k)}$ are the execution times of $g(x)$ and $g(x')$ evaluations respectively), thus t_{EXEC} increases when m is increased. On the other hand, if the two evaluations cannot be performed concurrently then the serial evaluation of each state point results in a total execution time of the search given by $t_{EXEC} = \sum_k (t_{EXEC,g(x_k)} + t_{EXEC,g(x'_k)})$. In this case, t_{EXEC} can be shortened by reducing NEVALS, hence a value of $m = m^*$ should undoubtedly be considered to approach a minimum in NEVALS for a random \mathbf{x}_0 .

6.7 One-dimensional noisy hill climbs

In the previous section we found that the NEVALS- m curves for threshold climbing runs exhibited a characteristic form with a dip for low m and an increasing tail for high m . But is this purely a result of performing hill-climbs on the yield landscape model \hat{Q} ? To confirm that the characteristic form of the attributes curves (NEVALS, NSTEPS, NTRIALS) observed are not exclusive to \hat{Q} , we performed similar threshold hill-climbing runs on simple 1-dimensional (1D) landscapes.

1D landscapes are easier to model and analyse than 2D landscapes. Restricted to moving along a single dimension forces a climber to move across a local maximum lying between the initial point and destination (threshold fitness), eliminating the possibility for it to go around the maximum in higher dimensions. All climbers thus consider the same available path to reaching the threshold fitness, restricting the changes in threshold climbing results as a consequence of the modification of the single path, rather than the contributions of a subset of available paths in 2D. This isolates the contribution of a local maximum to the threshold climbing results purely to the path concerning the crossing of the maximum. The runs were performed on an assortment of 1D landscapes to gain further insight into the interaction between the m parameter and noisy fitness landscape features, and whether the NEVALS minimum is always found at low m .

6.7.1 1D fitness landscapes model and experimental setup

Climbers move along a discretised x axis of integer values. A linear fitness function $f_{1D}(x)$ is defined by

$$f_{1D}(x) = \gamma x, \quad (6.10)$$

with γ the gradient of the fitness slope. Noise is added in a similar manner to our yield landscape, by appending a constant normal distribution $\mathcal{N}_{\text{norm}}(0, \sigma)$ with zero mean and standard deviation σ to each point x to give a noisy measured fitness $g_{1D}(x)$:

$$g_{1D}(x) = f_{1D}(x) + Y, \quad Y \sim \mathcal{N}_{\text{norm}}(0, \sigma). \quad (6.11)$$

We used $\gamma = 1$ in our studies.

For finite σ , climbing on g_{1D} is analogous to a biased, dampened 1D random walk. The ratio between the gradient γ and standard deviation σ determines the climbing speed. The ascent is sharper for $\sigma/\gamma < 1.0$, with no moves in the negative x direction if $\sigma = 0$. The influence of the gradient is suppressed when $\sigma/\gamma \gg 1$. The resulting hill climb exhibits a dampened unbiased random walk when $\sigma \rightarrow \infty$.

Aside from this base noisy fitness landscape g_{1D} (labelled G1), we considered a few other variants by either reducing the noise and/or adding local optima. Local optima were added by defining pairs of integer points x_a, x_b where the gradient of f_{1D} between (x_a, x_b) was reversed. For example, defining a negative gradient $-\gamma$ in the negative gradient interval (x_a, x_b) of size $l = x_b - x_a$, a piecewise 1D function with a single local maximum at point x_a is defined as

$$f_{\text{singlemaximum}}(x) = \begin{cases} \gamma x, & \text{for } x \leq x_a, \\ \gamma x_a - \gamma(x - x_a), & \text{if } x_a \leq x \leq x_b, \\ \gamma x - 2\gamma(x_b - x_a), & \text{if } x \geq x_b. \end{cases} \quad (6.12)$$

The basin of attraction for this local maximum is $x < x_b$. Once at x_a , a climber can escape the local maximum by moving downhill for l steps in the positive x direction. Local maxima were inserted in the range $0 < x < 30$. The effect of the position of (x_a, x_b) on the performance was not tested but is believed to not have a huge consequence on the normalised form. Plots of these underlying fitness functions F1-F6 can be found in Figure 6.14. Combined with a choice of σ and a neighbourhood $\mathcal{N}(x) = \{x - 1, x + 1\}$ yields the noisy versions of the landscapes G1 to G6. The fitness functions and landscapes used in this section are listed in Table 6.2.

A total of 4000 climbers with m ranging from 1 to 100 were initialised at $x = 0$ and allowed to walk on each of the landscapes G1 to G6 for at most $k_{\text{max}} = 1000$ iterations, or alternatively terminated once they reached a threshold fitness of $F^* = 70$. Again, the number of iterations **NSTEPS**, number of evaluations **NEVALS** and the number of trials **NTRIALS** were recorded upon termination. The results of these climbs are plotted in Figure 6.15, focusing on the change in climbing properties relative to the base case $m = 1$ for each landscape tested.

Table 6.2: The six 1D noisy fitness landscapes G1 to G6 tested, along with their respective fitness function F1-F6 and features. The noisy landscapes vary in the standard deviation σ of the normal distribution noise, and the number and size of negative gradient intervals (NGIs) in its fitness function F. Plots of F1 to F6 are found in Figure 6.14. The defining features of each landscape are: (G1) no maximum (equivalent to g_{1D} with $\sigma = 1$); (G2) small standard deviation (equivalent to g_{1D} with $\sigma = 0.001$); (G3) single maxima with reduced standard deviation; (G4) single maximum; (G5) two maxima; (G6) single large maximum.

| Noisy landscape | Fitness function | σ | Number of NGI | Size of NGI (l) |
|-----------------|------------------|----------|---------------|---------------------|
| G1 | F1 | 1 | 0 | — |
| G2 | F2 | 0.001 | 0 | — |
| G3 | F3 | 0.5 | 1 | 1 |
| G4 | F4 | 1 | 1 | 1 |
| G5 | F5 | 1 | 2 | 1 |
| G6 | F6 | 1 | 1 | 2 |

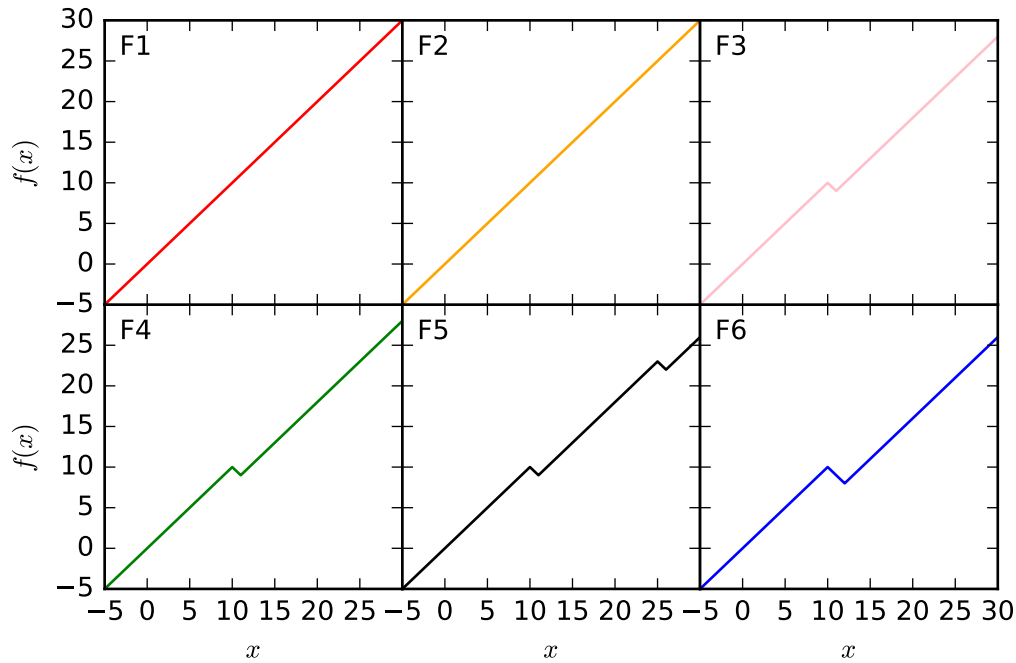


Figure 6.14: Plots of the six 1D fitness landscapes, F1 to F6, used in study. Noise is not shown (See Table 6.2). All fitness landscapes are linear beyond the x range shown.

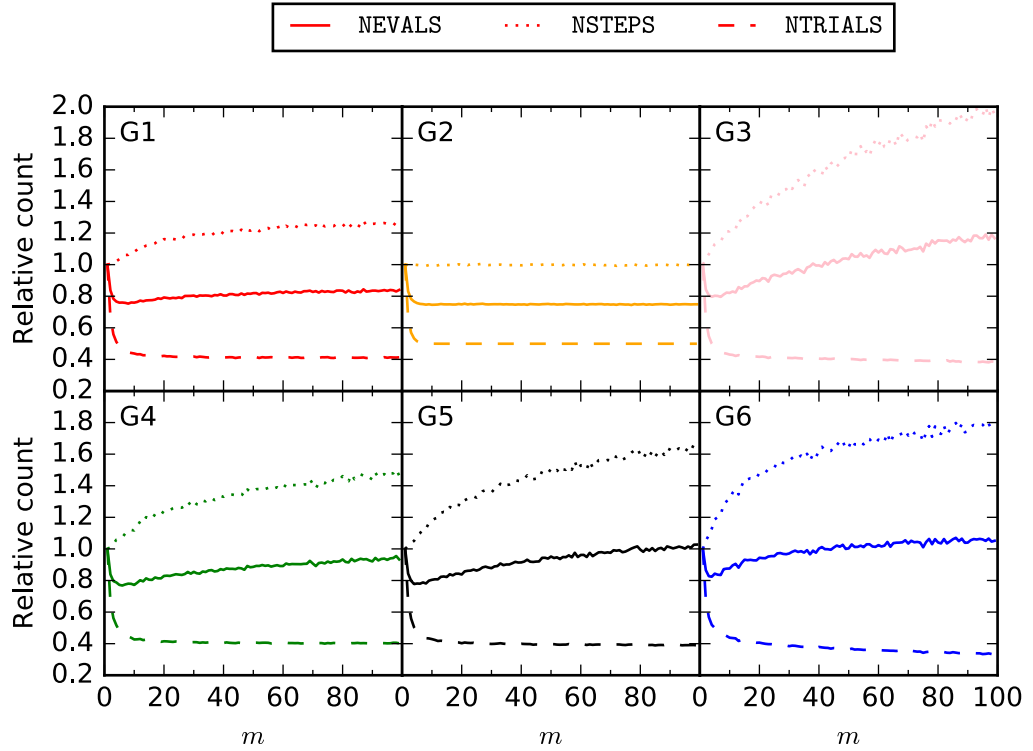


Figure 6.15: Plots of relative NEVALS and NSTEPS counts with respect $m = 1$ case, for the six different 1D fitness landscapes G1 to G6.

6.7.2 Results

For all landscapes G1 to G6, NTRIALS exhibited an almost identical asymptotic decrease to $\sim 40\%$ of the $m = 1$ case. NSTEPS increased with m at different rates depending on the landscape and hence the NEVALS tail also differed. The value of m^* was almost always less than 10, with associated NEVALS value greater than 76% (nearest integer).

The increase in NSTEPS as m is increased was negligible when $\sigma = 0.01$ (G2), but significant in the presence of noise $\sigma = 1$ (G1). However, the better signal-to-noise ratio did not bode well for landscapes with local maxima, with a larger increase in NSTEPS as m is increased for $\sigma = 0.5$ (G3) than for $\sigma = 1.0$ (G4). The gradient of the NSTEPS- m curve was more positive as the number of local maxima and/or noise increased, with the curve affected more by an increased prominence and size of local maxima (G6—bigger local maximum) than by an increase in the number of local maxima (G5—more local maxima). The non-monotonic NEVALS behaviour followed the NSTEPS trend, where a sharper NSTEPS increase led to a lower drop in NEVALS minimum and smaller m^* value.

6.7.3 Discussion

Even with a reduced dimensionality and minimal form of a noisy fitness landscape, the changes in threshold climbing behaviour as m is varied on the 1D noisy landscapes were found to be similar in form to that on the yield landscape model \hat{Q} (a 2D landscape) in the previous section. This strongly suggests that these changes in climbing behaviour are

universal across all noisy landscapes models. The **NTRIALS** curves are largely the same across the different landscapes, whereas **NSTEPS** at high m exhibited a high dependency on the local maxima present in the landscape.

Insight into why this is the case can be found by studying the effects of m on the move probabilities as discovered in Section 6.5. For simplicity, let us assume a linear noisy landscape f_{1D} with fixed noise, but also that a current fitness evaluation $g(x)$ at point x will always yield a fixed fluctuation c ; neighbouring evaluations are still noisy. The probability of a successful move p_c will thus be the same for every iteration k . Moreover, let us assume that a threshold climb requires **NSUCCS** number of successful moves to reach threshold fitness for any value of m .

The probability of a successful trial (a trial ending with a successful move before it reaches m) is

$$h_x(m) = 1 - (1 - p_c)^m, \quad (6.13)$$

(see Eqn. 6.9) and increases as m is increased. The expected number of trials required to observe a successful trial is

$$\mathbb{E}[\text{successful trial}] = \frac{1}{h_x(m)}. \quad (6.14)$$

$\mathbb{E}[\text{successful trial}]$ decreases as m is increased. It follows that **NTRIALS** is proportional to $\mathbb{E}[\text{successful trial}] \times \text{NSUCCS}$.

Meanwhile, the expected number of iterations per trial is the geometric sum

$$\mathbb{E}[\text{iteration per trial}] = \sum_{j=1}^m (1 - p_c)^{j-1} p_c j, \quad (6.15)$$

with the number of iterations to reach threshold fitness **NSTEPS** = $\mathbb{E}[\text{iteration per trial}] \times \text{NTRIALS}$. As m appears in the summation of $\mathbb{E}[\text{iteration per trial}]$, **NSTEPS** increases as m is increased. The sum of decreasing **NTRIALS** and increasing **NSTEPS** yields the non-monotonic **NEVALS** as seen in the climbs. This describes the geometric decay of **NTRIALS** contrasting with the increasing **NSTEPS** as m is increased. Since these hold for all fluctuations c , and each p_c is independent random variable for each iteration, the behaviour is expected to be of similar form in the noisy current fitness evaluation case.

The presence of local maxima and different noises for landscapes G2 to G6 alter the average move probabilities. The expected number of iterations required to move beyond a point x is increased when a local maxima is introduced at x . One possible explanation for this is that $\mathbb{E}[\text{iteration per trial}]$ is increased towards the limit m , while the number of trials increases slightly, resulting in a large multiplicative increase in **NSTEPS**.

In summary, simulated threshold hill-climbing runs on 1D noisy landscapes and observed similar trends to those on \hat{Q} , confirming the consistent improvement in **NEVALS** from increasing m . The brief insight into the expected move outcomes given above showed that this has a tendency to increase the number of iterations while reducing the number of trials. In two or more dimensions the ability to take locally different paths offers different routes to threshold fitness (leading to drastically different **NSUCCS**). Further investigation

is required to confirm how this influences the trends discussed here.

6.8 Revisiting the climbing schedule strategy: searching a changing landscape

As noted in Section 4.5, the yield landscape Q is time-dependent. The region of significant assembly is established at early times and stays relevant throughout the duration of the assembly. One strategy for cutting down on execution times and computational resources is to deploy a climbing schedule to tune the parameter values for early-time assembly yield so that shorter simulations are executed initially, before gradually considering the yield at later durations and eventually at final duration t_f . This was used in the preliminary study in Section 6.1. In this section, we test the performance of the modified hill-climbing algorithm (Algorithm 3) with this climbing schedule strategy on the yield landscape model \hat{Q} for different m . The climbing schedule trigger in Section 6.1 is not considered here. There are two aims for implementing the climbing schedule: one is to reduce the computational time and thus increase the efficiency; the other is to assist climbers in converging towards higher fitness and thus increase their robustness.

6.8.1 Setup

To model the climbing schedule using the yield landscape model \hat{Q} , we performed hill-climbing runs on a multi-duration yield landscape \hat{Q}_{mult} consisting of a set of noisy yield landscapes models $\hat{Q}_{mult} = \{\hat{Q}_{t_1}, \hat{Q}_{t_2}, \dots, \hat{Q}_{t_f}\}$ for a number of self-assembly durations $t_i = t_V / (\text{VMMC cycles})$, with a schedule update rate κ . A hill-climbing run is performed on \hat{Q}_{mult} with a climber starting on \hat{Q}_{t_1} . After κ iterations, the climber advances onto \hat{Q}_{t_2} , preserving its position \mathbf{x}_κ , and is prompted to trial its current yield. The climber continues its climb on the new landscape for κ steps before repeating the process to larger durations t_i , until it reaches \hat{Q}_{t_f} where it climbs until it exhausts the remainder of its allocated iterations k_{\max} , i.e. the termination condition is the maximum climbing length k_{\max} . We used $\{t_1, t_2, t_3, t_f\} = \{10^6, 5 \times 10^6, 10^7, 2 \times 10^7\}$ as our experimental set of durations and $\kappa \in \{10, 20, 50, 80\}$ for the schedule update rates on each landscape. The average \bar{Q} trajectory were tracked for multi-duration hill-climbing runs performed on \hat{Q}_{mult} and on a single final time landscape \hat{Q}_{t_f} for comparison (labelled $\kappa = 0$) for the modified hill-climbing algorithm (Algorithm 3) with protocol **S1a** and $m = 1, 8$ and 1000.

A couple of intuitive guidelines concerning the number and distribution of durations were followed when choosing $\{t_i\}$. A larger \hat{Q}_{mult} set systematically extends the number of iterations before a climber arrives at \hat{Q}_{t_f} —we call this the *queue time*. The queue time is also influenced by the choice of κ : a climber takes $\kappa|\hat{Q}_{mult}|$ iterations to access the t_f landscape. The termination condition k_{\max} defines an upper bound for κ and $|\hat{Q}_{mult}|$, since if $\kappa(|\hat{Q}_{mult}| - 1) > k_{\max}$ then a climber would fail to reach \hat{Q}_{t_f} . We chose a set of four durations (including t_f) to keep the number of landscapes $|\hat{Q}_{mult}|$ low. To further ensure that a climber does not neglect climbing the final \hat{Q}_{t_f} landscape, we imposed the criterion that climbers had to climb the \hat{Q}_{t_f} for a minimum of κ iterations, i.e. $1 \leq \kappa \leq$

$(k_{\max}/|\hat{Q}_{mult}|)$. A selection of durations $\{t_i\}$ with a sensible distribution is also helpful to observe sufficient change between landscapes \hat{Q}_{t_i} . We chose $\{t_i\}$ such that they were closer for the early stages, but doubled between the penultimate and final durations i.e. $t_f = 2t_3$ to complement the logistic yield growth.

6.8.2 Results

The observed outcomes for k -dependent \bar{Q} trajectories between \hat{Q}_{t_f} -only climbs ($\kappa = 0$) and \hat{Q}_{mult} climbs (positive κ) were dependent on m . For low m (e.g. $m = 1, 8$), similar terminal \bar{Q} were seen for all κ values, regardless of the initial point \mathbf{x}_0 . Positive κ climbs exhibited a slow start as expected before joining the \bar{Q} trajectory of $\kappa = 0$ at larger k , subject to the queue time. Examples shown include climbs initialised at $\mathbf{x}_0 = (6, 10)$ (near secondary peak, Fig. 6.16a) and $\mathbf{x}_0 = (14, 19)$ (in plateau far away from peak, Fig. 6.16b). For $m = 1000$, significant improvements in the terminal \bar{Q} were observed. Climbers reached higher \bar{Q} but did so at larger k as κ was increased, due in part to the queue time.

Of course, our motivation for the climbing schedule is to reduce t_{TOTCPU} of the search. The same \bar{Q} trajectories in Fig. 6.16 are shown in Fig. 6.17 but plotted against t_{TOTCPU} . The computational times used are an approximation of the actual simulations. Only the full execution time $t_{EXEC,sim}$ of each single-CPU VMMC simulation was recorded, so we assumed a linear relation between the simulation duration t_V and the mean execution time $\langle t_{EXEC,sim} \rangle$ for each \mathbf{x} to infer the execution times of the simulations at different t_i . As argued before, VMMC simulations become slower as clusters grow hence this linear relation overestimates the computational demand of simulations terminated at small t_V . In general, for a given m value, climbers reached similar or higher \bar{Q} at lower t_{TOTCPU} as κ was increased, with the effect most apparent for $m = 1000$.

The observed changes in terminal fitness and t_{TOTCPU} as κ increased were dependent on the choice of the initial point and were not guaranteed to provide better self-assembly parameters at termination. However, considering all the initial points tested, the climbing schedule does provide some benefits to the hill-climbing outcomes. In general, as κ increased, the terminal \bar{Q} stayed similar for small m but saw a relatively large increase for large m , with some cases showing an increase of up to triple the original κ value (Fig. 6.18a). These increases are not necessarily monotonic however. Overall this induces an improvement in the robustness of the hill-climbing, since the worst-case outcome (terminal \bar{Q}) is not as heavily reliant on \mathbf{x}_0 . This comes as a bonus on top of the lower t_{TOTCPU} (and thus lower t_{EXEC}) required to perform such fixed k_{max} climbs. Low m saw the largest proportional reduction in t_{TOTCPU} , in some cases by over a quarter of the original $\kappa = 0$ climbs when $\kappa = 80$.

6.8.3 Discussion

The climbing schedule improves terminal \bar{Q} by helping climbers handle local maxima and long simulation durations at the same \mathbf{x} . A climber avoids prominent late t_V local maxima by traversing the region at early t_V . For example, in Fig. 6.16a, positive κ climbers avoided

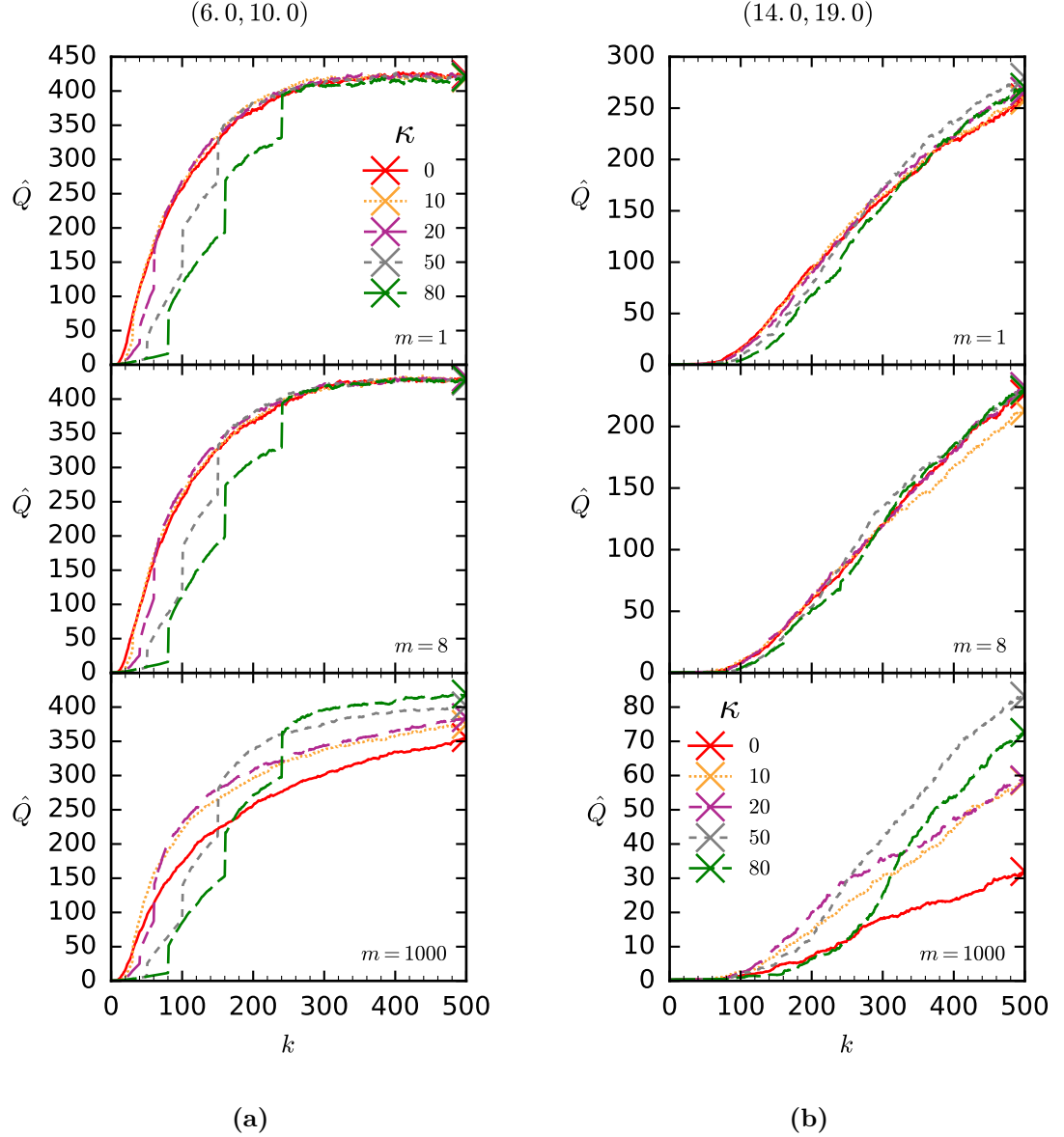


Figure 6.16: Plots comparing the expected true fitness \bar{Q} trajectories, dependent on iteration k , for hill-climbing runs at different m and κ , initialised (a) in the liquid phase (6, 10) and (b) at a strongly interacting (10, 19) state. Each plot, grouped by m , shows the mean fitness trajectory of a set of climbers on $\hat{Q}(t_f = 2 \times 10^7)$ (labelled $\kappa = 0$) alongside the mean fitness trajectories of climbers searching on the multi-duration yield landscape $\hat{Q}_{mult} = \{\hat{Q}(t_i)\}$ with $t_i \in \{10^6, 5 \times 10^6, 10^7, 2 \times 10^7\}$. Advancing in increasing order, each climbers spent κ steps at each intermediate landscape, before it spent the rest of the allocated iterations ($k_{\max} = 500$) climbing $\hat{Q}(t_f)$. Error bars are smaller than the marker size.

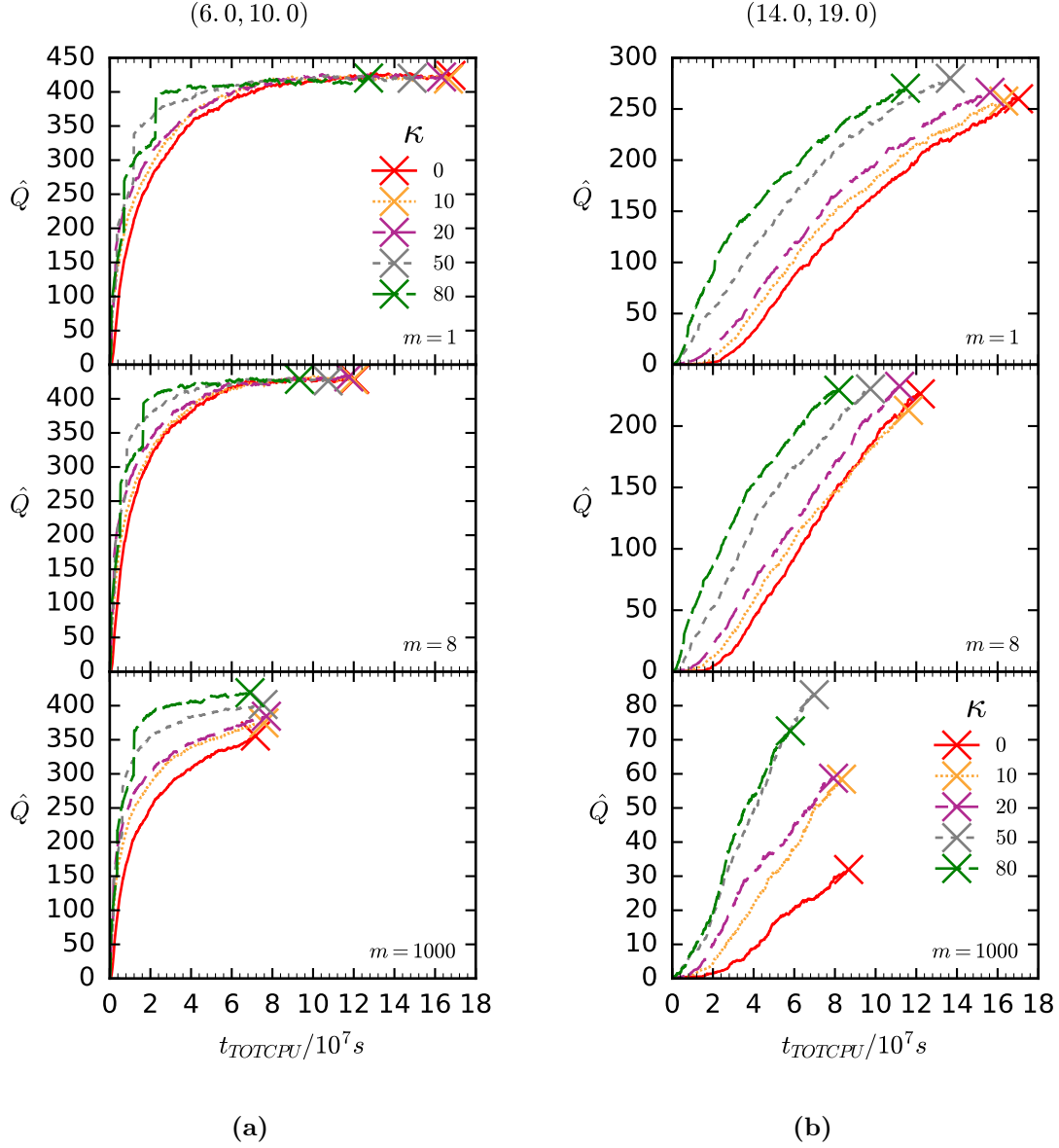
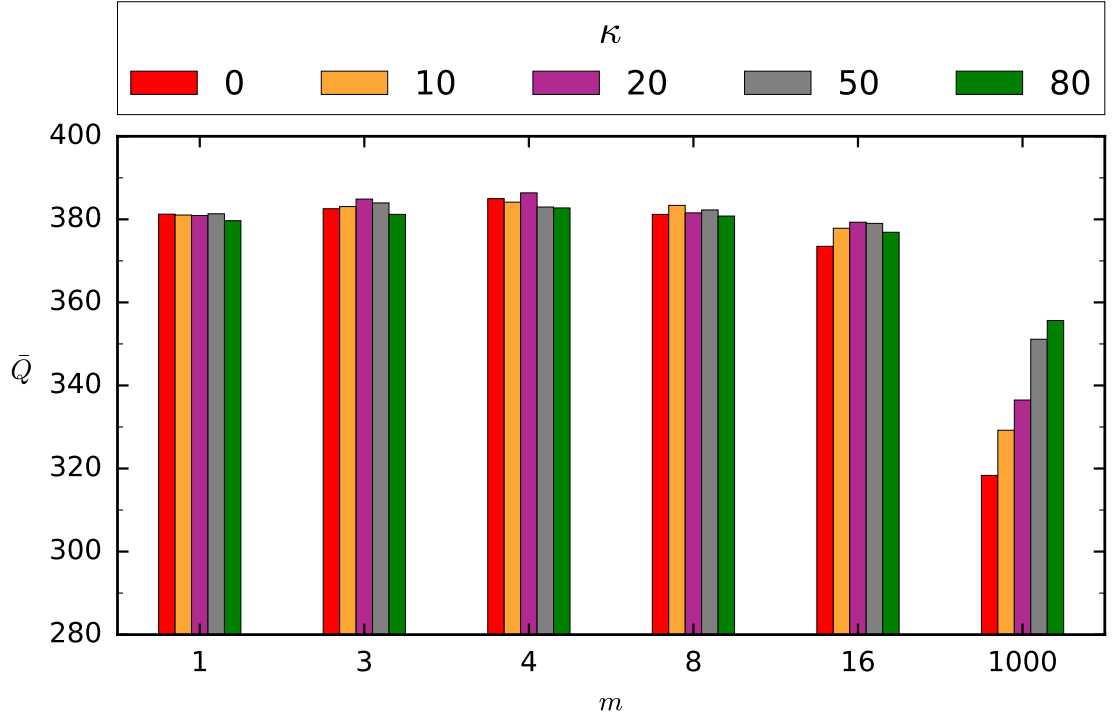
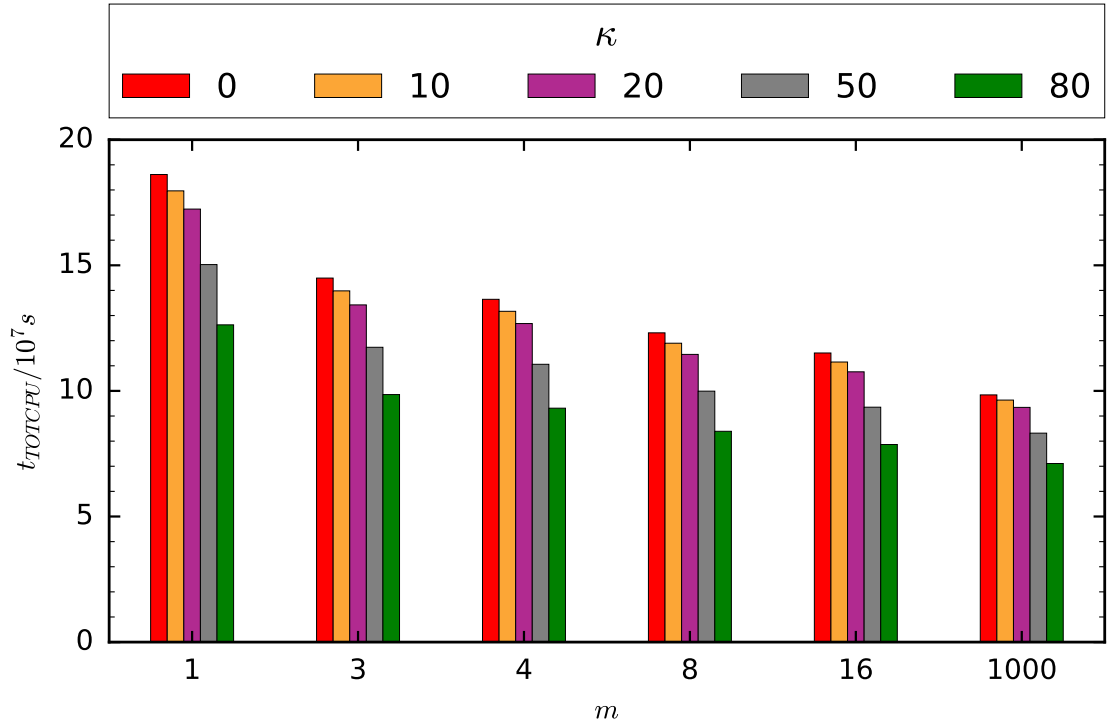


Figure 6.17: Plots comparing the expected fitnesses for hill-climbing runs at different m and κ , initialised (a) in the liquid phase (6,10) and (b) at a strongly interacting (10,19) state. Each plot, grouped by m , shows the mean fitness trajectory of a set of climbers on $\hat{Q}(t_f = 2 \times 10^7)$ (labelled $\kappa = 0$), compared to the mean fitness trajectories of climbers searching on the multi-duration yield landscape $\hat{Q}_{mult} = \{\hat{Q}(t_V)\}$ with $t_V \in \{10^6, 5 \times 10^6, 10^7, 2 \times 10^7\}$. Advancing in increasing order, climbers spend κ steps at each intermediate landscape, before spending the rest of the allocated iterations ($k_{max} = 500$) climbing $\hat{Q}(t_f)$. Error bars are smaller than the marker size.



(a)



(b)

Figure 6.18: Plots of (a) underlying average fitness \bar{Q} and (b) total CPU cost t_{TOTCPU} at termination $k_{\max} = 500$ for different κ and m averaged over all initial points \mathbf{x}_0 sampled (500 hill climbs each).

the secondary peak at (7.2,11) which only fully formed after 7×10^6 VMMC cycles. A larger κ allowed time for a climber to reach the global peak region at lower w before this secondary peak formed, eliminating the need to waste iterations trying to escape this prominent local maxima. Climbers initialised in the low- ε high- w region, where a ridged surface developed at late t_V , also showed significant improvements when climbing on \hat{Q}_{mult} instead.

The schedule forces a retrieval at every advancement of the landscape, assisting climbers with \hat{Q} with highly positive fluctuation Y to move before the trial period limit m is reached. Together, the climbing schedule helps climbers with limited explorative power escape local maxima, which explains why a greater effect is observed for $m = 1000$ than for $m = 1$ or 8.

In terms of computational demand, the climbing schedule for all κ successfully reduced the potential t_{TOTCPU} demand. In a few cases an increase in t_{TOTCPU} was observed when switching from $\kappa = 0$ to positive κ (e.g. $m = 1000$ in Fig. 6.16a) due the longer $\langle t_{EXEC,sim} \rangle$ as higher \bar{Q} was reached. Moreover, the execution time t_{EXEC} of a hill-climbing run adopting a climbing schedule would also show similar reduction, addressing the issues of long t_{EXEC} due to the serial nature of the hill-climbing search.

We did not test $\kappa = 125$, the upper bound of the κ criterion. Based on the results of the κ values tested, $\kappa = 125$ has the potential to offer the most reliable improvement in both \bar{Q} reached and t_{TOTCPU} taken, leading to an even more robust parametrised hill-climbing algorithm. From the trajectory plots in Fig. 6.17, threshold climbs would also benefit from the climbing schedule: a lower t_{TOTCPU} can be sufficient for a climber to reach a given threshold value. Moreover, only constant κ values were tested. Recall from Sec. 6.1.1 the intuition behind the use of the climbing schedule: by eliminating low yielding regions via searching the assembly yield at earlier durations, climbers spend a fraction of the cost to find the hill (basin of attraction) before targeting the global peak at the final duration t_f . Starting from a large value, a decreasing κ should be considered to allocate more iterations for low t_i exploration, further reducing t_{TOTCPU} . This assumes that the high yielding region of a yield landscape does not change drastically throughout the assembly.

The climbing schedule can be viewed as a numerical extension to the preliminary determination of the scope of parameter space, where the search range of parameter values is identified. The elimination of low yielding regions of parameter space at early durations proves effective in not only reducing the computational demand and execution time of the search but also in producing a more reliable search outcome.

Contrary to the argument for reducing the size of a problem by consolidating to a single objective, a multiobjective optimisation technique may enable a faster hill-climbing search of the target global yield peak by giving climbers a better chance of starting their climb near significant yield solutions when they eventually search at the final duration t_f . A ‘quick-and-dirty’ approach to cut down on computational demand and execution time would be to identify the objectives that would converge at early times and be consistent throughout the assembly. For our definition of Q , the structural quality $\chi_{6,C}/N_C$ was found to converge within 10^5 VMMC cycles. For targeting the honeycomb structure, a climber can be designed to climb the $\chi_{6,C}/N_C$ landscape at early times, consistent

throughout most of the assembly, before switching to climbing the full Q landscape once the parameters that produce locally ordered structures are found.

6.9 Conclusion

We have shown several stochastic hill-climbing metaheuristic protocols and strategies for finding parameters yielding (near-)optimal assembly on noisy yield landscapes. These protocols and strategies (current fitness trial period with limit m ; cumulative averaging; multi-duration climbing schedule) were studied to provide approaches for reducing the computational cost of searches. The reduction of computational resources for threshold climbs were shown to be possible at low m values without negatively affecting the search outcome, addressing one of the questions in the chapter introduction, but required more iterations to be executed. Increasing r_{hist} in cumulative averaging was demonstrated to be a viable replacement to increasing r in fitness averaging, but consequences of correlated fitness evaluations values were not investigated. The multi-duration climbing schedule offered multiple advantages: it reduced both computational and execution times of searches, and helped climbers escape/avoid local maxima to reach higher terminal fitness.

These three methods for improving hill-climbing search on noisy yield landscapes are independent of each other. A combination of all three methods was not investigated. The results shown here represent a fraction of the possible parametric combinations of the methods, highlighting the possible customisations for tailoring to a yield landscape (or indeed any noisy fitness landscape) where expensive simulations are of concern. Furthermore, they suggest that a scheduled implementation of m and r_{hist} , much like that of r in fitness averaging, is worth investigating.

For self-assembly parameter tuning, where evaluations are expensive, the global peak region seems to be an acceptable default target, and given the effect of noise, should be considered as the most cost-effective aim, rather than finding the global maximum. Efforts to find good initial points is encouraged. Reducing noise in the latter part of a hill-climbing search is key to converging to a local maxima, if not the global maximum. Although we have shown how the metaheuristic protocols introduced in this thesis are able to handle and reduce noise, the diminishing returns of these strategies as climbers reach higher fitness lead to long t_{EXEC} and prevent them from making hill-climbing a practical technique to investigate self-assembly, as computational resources are in abundance compared to time. It seems that fitness averaging over repeated sampling (increasing the evaluation sample size r), which we assumed to be a costly strategy at the start, is at least part of the key to solving this issue. By including this, a hill-climbing search to find high-yielding parameter values would take advantage of the reduced computational cost required to reach the global peak region (by choosing a suitable m and r_{hist} , and implementing a climbing schedule), before increasing r upon reaching the global peak region to exploit concurrency (running multiple simulations per iteration to search more accurately), which should reduce the t_{EXEC} times seen thus far. This may enable hill-climbing to become the technique of choice over brute-force scan for computational self-assembly parameter tuning.

Our results are intended to be used as a basic guide for informing future selection and application of these search methods, whether used alone or in conjunction with other meta-heuristic techniques, to diversify/intensify the simple stochastic hill-climbing algorithm on noisy yield landscapes.

Chapter 7

Outlook

The aims of this study were directed toward improving the parameter tuning methods in forward design of colloids, where a brute-force scan of parameter space is the conventional method for identifying high yielding parameter values. We set out to empirically test the stochastic hill-climbing heuristic as an alternative approach for tuning values at a reduced computational cost and execution time, and assess additional search protocols to further this goal.

Our self-assembly example problem targeted the values of the interaction strength ε and patch flexibility w of a three-patched disk model that would maximise the yield of large, compact, circular honeycomb cluster aggregates. By constructing an appropriate yield measure Q involving the shape factor we extended the scope of targeted design criteria from conventional microstructure and size based measures to ones targeting the macrostructure. A systematic scan revealed the dependency of the assembly outcome on ε and w , along with the fluctuations in Q which can be interpreted as noise on the resulting yield landscape. The noisy yield measurements also indicate the reproducibility of assembled structures, an important factor to consider when designing systems that will assemble readily into the desired state. Despite the different levels of cluster-cluster agglomeration producing inconsistent assembly outcomes across the parameter space, optimal (highest yielding) assembly was identified in a region where the system exhibited steady growth of a single cluster.

The assessment of the stochastic hill-climbing heuristic was undertaken on the time-dependent noisy yield landscape model \hat{Q} , constructed to bypass the costly particle simulations. In terms of climbing ability, we found that the standard stochastic hill-climbing algorithm is a reliable way for locating the global peak region of the noisy yield landscape, reaching it consistently from different initial points. The results highlight the benefits of providing a good initial point near to the global peak, so effort into identifying a good range for the parameter space is encouraged. Several metaheuristic aspects and protocols were considered on top to further improve its effectiveness.

The concept of a trial period and a trial period limit m were initially introduced to encourage the reuse of fitness values obtained from expensive evaluations. Surprisingly this ‘investment-free’ parameter showed the ability to intensify a search while reducing the number of simulations performed per iteration. An assessment of the fluctuation depen-

dent move probabilities showed that increasing m discourages the climber from performing incorrect (downhill) moves, resulting in a proportional rise of correct uphill moves at the cost of slowing down the climber’s exploration. This influenced a climber’s ability to escape local maxima that led to improvements in the terminal yield fitness of the climb in certain cases.

However m alone was not enough to counter the large spread in the terminal position of climbers in the global peak region stemming from the large noise. The cumulative averaging protocol—another ‘investment-free’ protocol emulating the reduced noise effects offered by increasing the evaluation sample size r —addresses this aspect, enabling climbers to converge to a highly localised region once the correct basin of attraction was found. However, the stronger intensification effects, controlled by the history sample size r_{\max} , deteriorate the climbing performance significantly if a climber is initialised far away from the peak region. We expect a scheduled implementation of both of these intensification protocols, gradually increasing m and r_{\max} based on k , will produce a reliable ascent yet localised convergence to optimal yield.

Regarding the computation cost t_{TOTCPU} of the search, the regular stochastic hill-climbing algorithm consistently reached the global peak region within 300 iterations depending on the initial point. This is superior to the brute-force scan which evaluates all 920 state points in the parameter space at least once. On the other hand, a major limitation of the hill-climbing search lies in its execution time t_{EXEC} , which is of similar order to t_{TOTCPU} due to the serial nature of the algorithm, strongly disfavours its usage. The adoption of the hill-climbing search is therefore subject to the parallelisability of the brute-force scan. If the stochastic hill-climbing algorithm is favoured however, then the results from the threshold climbing studies show that it is almost always worth having $m \geq 2$, but only up to small integers. The catch is that there is a trade off between the NEVALS, total number of evaluations called, and NSTEPS, the number of iterations executed, affecting t_{TOTCPU} and t_{EXEC} respectively.

A break down of the threshold climb results as a function of m reveals that the rapid initial drop in NTRIALS, the expected number of trials performed, balanced by the landscape-sensitive increase in NSTEPS, is responsible for the non-monotonic behaviour NEVALS with reduction at low m . This reduction from increasing m weakens in effectiveness thereafter and can even be undesirable when a landscape is too rough and its signal-to-noise ratio is too low as m cannot overcome low initial move probabilities, leading to an increase in climbs that would take impractically long times to reach threshold fitness. The selection of a well-behaved yield measure is therefore critical to the enhancement brought by tuning m . Threshold climbs to a target value also demonstrate a termination criterion for searching for a ‘good-enough’ solution. This may be the most practical termination condition for hill-climbing on noisy landscapes since climbers will never stop at the local optimum point but continue to move and explore the parameter space indefinitely.

The iterative nature of hill-climbing hinders the algorithm’s applicability as the preferred standalone search method as execution times are too long. Its ability to reduce computational demand is best exploited when running simultaneous hill-climbing runs to incorporate concurrency in a different way to brute-force. Otherwise, it is best used as a

preliminary search technique for eliminating poor yielding regions, before a more thorough search is applied. Looking beyond the metaheuristic strategies studied here, increasing the evaluation sample size r to independently reduce noise quickly after a climber reaches the global peak region is crucial to lowering the execution time and thus increasing the search efficiency even further.

Our final strategy considered is multi-duration hill-climbing. This strategy, taking advantage of the time dependency of the yield landscape, is beneficial for multiple reasons. Firstly, it cuts down on the duration of the particle simulations at early iterations, directly reducing the long t_{EXEC} of hill-climbing. Moreover, it provides a ‘nudge’ for the climber every time it progresses to longer durations, allowing climbers that are stuck to escape local maxima. This provides a good complementary strategy to aid the trial period limit and cumulative averaging strategies, whose powers lie in the intensification of the climbing behaviour, helping climbers with high m or large r_{max} to move when they get stuck.

In conclusion, the stochastic hill-climbing algorithm and associated protocols serve to aid in the exploration of dynamical colloidal self-assembly design parameters. Although the results were empirically derived from observations on the noisy yield landscape model \hat{Q} , they summarise underlying mechanisms that manifest due to general characteristics possessed by all yield landscapes. Our results suggest that improvements to the conventional brute-force scan are not particularly significant and subject to several restrictions, but it nevertheless expands the tools available for parameter tuning. It is fundamentally down to the user to determine whether the hill-climbing algorithm is suitable based on the optimal management of simulations and the available computational resources.

7.1 Future Work

Our yield landscape model is based on the yield data gathered from the particle simulations, limiting our hill-climbing search studies to a fixed step size. Varying the step size (i.e. having a variable neighbourhood) of a local search is a common method for allowing climbers to respond to the changing terrain of the local landscape. This could potentially lead to significant improvements in the performance of hill-climbing on these yield landscapes as they typically only have a small region of high yielding parameter values. It is not obvious how the handling of noisy yield measurements and the protocol for changing the step size should proceed, but, as proven on many occasions, the opportunity for downhill moves may lead to enhanced performance of the heuristics.

One major quality of metaheuristics is its ability to optimise search efficiency via a metaheuristic variable, either by predetermining it using knowledge of the landscape or tuning it on-the-fly. As mentioned in Chapter 5.4.4, this is part of the current research of the evaluation sample size r in fitness averaging. In the same manner, an on-the-fly protocol for altering m would enable greater control of the climbing behaviour. Unfortunately the effect of increasing m is not trivially dependent on the noise variance and it is unknown whether a mathematical approach can provide further insight into this endeavour.

This does not prevent us from exploring simple variable m protocols that may help a climber adapt more effectively on a standard fitness landscape as opposed to a constant

m throughout. An idea for a variable m protocol would work in a similar fashion to the acceleration scheme of a non-gradient-based local search optimisation [135]. In this acceleration scheme, a successful move on a deterministic landscape is rewarded with a following move attempt in the same direction but at a larger step, increasing the speed (distance travelled over each unit iteration) of the climber. Successive moves of this type "accelerates" the speed of the search in the direction of travel until a move fails, upon which the speed is reset and the next iteration proceeds with a random hill-climbing move attempt.

The idea for a similar scheme involves increasing or decreasing the value of m based on whether a successful move was attempted during a trial before m was reached. The trial period hitting m successively could indicate the climber getting stuck at a local maxima, so more explorative power can be regained by slowly reducing m , down to a minimum of $m = 2$. If a successive move is completed before reaching the trial period limit then m is reset to its default value. While this default value would be subject to the landscape, from our studies we know that it should be kept relatively low. A more comprehensive study of the trial period could offer conditions on when m could be increased otherwise.

Ultimately, this study of stochastic hill-climbing search and strategies was undertaken considering its implementation to the design and tuning of colloidal self-assembly parameters. We limited ourselves to optimising two interaction parameters. A consideration of multiple interaction parameters is required as colloidal designs begin to move away from simple designs. For finite low dimensions, we expect the effect of m on the threshold climbing results to be similar under a basic neighbourhood definition. However, further investigation is required to understand whether this still holds in higher dimensions, where preferred local trajectories for climbers may be plentiful when in large neighbourhoods and the number of local optima decreases [154] (more directions are available to escape into). It is unclear how this changes the effect of increasing m on threshold climbs, and is likely heavily dependent on the landscape. One plausible trend is a dampening of the effects of m on the climbing behaviour seen in our studies as the number of dimensions increase, as more paths are available to go from point a to point b , hence the reduction in **NTRIALS** weakens. Another entirely different possibility is that increasing m has negligible effect on **NSTEPS** while **NTRIALS** is reduced, hence the increasing tail in **NEVALS** disappears. The use of a more complex neighbourhood definition may lead to improvements in the parameter tuning in general [145; 155] and in relation to m . Research into the effects of neighbourhood definitions and the coupling of different design parameters will likely offer further insight into standard procedures for creating complementary neighbourhood definitions for more efficient exploration of yield landscapes.

Although we have limited our application to forward design, the general optimisation framework adopted here means that our results are relevant to any applications of iterative local-search algorithms on noisy landscapes where evaluation of the current solution is performed at every iteration.

Bibliography

- [1] G. A. Ozin, K. Hou, B. V. Lotsch, L. Cademartiri, D. P. Puzzo, F. Scotognella, A. Ghadimi, J. Thomson, Nanofabrication by self-assembly, *Materials Today* 12 (5) 12–23 (2009).
- [2] S. Friedensen, J. T. Mlack, M. Drndić, Materials analysis and focused ion beam nanofabrication of topological insulator Bi₂Se₃, *Scientific Reports* 7 (1) 13466 (2017).
- [3] C. M. A. Parlett, M. A. Isaacs, S. K. Beaumont, L. M. Bingham, N. S. Hondow, K. Wilson, A. F. Lee, Spatially orthogonal chemical functionalization of a hierarchical pore network for catalytic cascade reactions, *Nature Materials* 15 178–182 (2015).
- [4] J. Wang, Q. Li, W. Knoll, U. Jonas, Preparation of multilayered trimodal colloid crystals and binary inverse opals, *J. Am. Chem. Soc.* 128 (49) 15606–15607 (2006).
- [5] J. D. Forster, J.-G. Park, M. Mittal, H. Noh, C. F. Schreck, C. S. O’Hern, H. Cao, E. M. Furst, E. R. Dufresne, Assembly of optical-scale dumbbells into dense photonic crystals, *ACS Nano* 5 (8) 6695–6700 (2011).
- [6] J. B. Matson, C. J. Newcomb, R. Bitton, S. I. Stupp, Nanostructure-templated control of drug release from peptide amphiphile nanofiber gels, *Soft Matter* 8 3586–3595 (2012).
- [7] M. J. Webber, J. B. Matson, V. K. Tamboli, S. I. Stupp, Controlled release of dexamethasone from peptide nanofiber gels to modulate inflammatory response, *Biomaterials* 33 (28) 6823–6832 (2012).
- [8] H. Cui, M. J. Webber, S. I. Stupp, Self-assembly of peptide amphiphiles: From molecules to nanostructures to biomaterials, *Peptide Science* 94 (1) 1–18 (2010).
- [9] G. van Anders, N. K. Ahmed, R. Smith, M. Engel, S. C. Glotzer, Entropically Patchy Particles: Engineering Valence through Shape Entropy, *ACS Nano* 8 (1) 931–940 (2014).
- [10] G. van Anders, D. Klotsa, N. K. Ahmed, M. Engel, S. C. Glotzer, Understanding shape entropy through local dense packing, *PNAS* 111 4812–4821 (2014).

- [11] X. Mao, Q. Chen, S. Granick, Entropy favours open colloidal lattices, *Nature Materials* 12 (3) 217–222 (2013).
- [12] C. J. Roberts, M. A. Blanco, Role of Anisotropic Interactions for Proteins and Patchy Nanoparticles, *J. Phys. Chem. B* 118 (44) 12599–12611 (2014).
- [13] S. Whitelam, I. Tamblyn, T. K. Haxton, M. B. Wieland, N. R. Champness, J. P. Garrahan, P. H. Beton, Common Physical Framework Explains Phase Behavior and Dynamics of Atomic, Molecular, and Polymeric Network Formers, *Phys. Rev. X* 4 011044 (2014).
- [14] A. Reinhardt, F. Romano, J. P. K. Doye, Computing Phase Diagrams for a Quasicrystal-Forming Patchy-Particle System, *Phys. Rev. Lett.* 110 255503 (2013).
- [15] M. J. Solomon, Directions for targeted self-assembly of anisotropic colloids from statistical thermodynamics, *Current Opinion in Colloid & Interface Science* 16 (2) 158–167 (2011).
- [16] S.-H. Kim, J.-M. Lim, S.-K. Lee, C.-J. Heo, S.-M. Yang, Biofunctional colloids and their assemblies, *Soft Matter* 6 1092–1110 (2010).
- [17] M. H. Lash, M. V. Fedorchak, J. J. McCarthy, S. R. Little, Scaling up self-assembly: bottom-up approaches to macroscopic particle organization, *Soft Matter* 11 5597–5609 (2015).
- [18] A. Martinez, M. Ravnik, B. Lucero, R. Visvanathan, S. Žumer, I. I. Smalyukh, Mutually tangled colloidal knots and induced defect loops in nematic fields, *Nature Materials* 13 258–263 (2014).
- [19] M. R. Jones, R. J. Macfarlane, B. Lee, J. Zhang, K. L. Young, A. J. Senesi, C. A. Mirkin, DNA-nanoparticle superlattices formed from anisotropic building blocks, *Nature Materials* 9 913–917 (2010).
- [20] F. Li, W. C. Yoo, M. B. Beernink, A. Stein, Site-Specific Functionalization of Anisotropic Nanoparticles: From Colloidal Atoms to Colloidal Molecules, *J. Am. Chem. Soc.* 131 (51) 18548–18555 (2009).
- [21] R. B. Jadrich, J. A. Bollinger, B. A. Lindquist, T. M. Truskett, Equilibrium cluster fluids: pair interactions via inverse design, *Soft Matter* 11 9342–9354 (2015).
- [22] B. A. Lindquist, R. B. Jadrich, T. M. Truskett, Assembly of nothing: equilibrium fluids with designed structured porosity, *Soft Matter* 12 2663–2667 (2016).
- [23] R. B. Jadrich, B. A. Lindquist, T. M. Truskett, Probabilistic inverse design for self-assembling materials, *J. Chem. Phys.* 146 (18) 184103 (2017).
- [24] B. Srinivasan, T. Vo, Y. Zhang, O. Gang, S. Kumar, V. Venkatasubramanian, Designing DNA-grafted particles that self-assemble into desired crystalline structures using the genetic algorithm, *PNAS* 110 (46) 18431–18435 (2013).

- [25] A. W. Long, A. L. Ferguson, Rational design of patchy colloids via landscape engineering, *Mol. Syst. Des. Eng.* 3 49–65 (2018).
- [26] A. J. Williamson, A. W. Wilber, J. P. K. Doye, A. A. Louis, Templated self-assembly of patchy particles, *Soft Matter* 7 3423–3431 (2011).
- [27] S. Whitelam, Hierarchical assembly may be a way to make large information-rich structures, *Soft Matter* 11 8225–8235 (2015).
- [28] S. Whitelam, E. H. Feng, M. F. Hagan, P. L. Geissler, The role of collective motion in examples of coarsening and self-assembly, *Soft Matter* 5 1251–1262 (2009).
- [29] E. Klavins, Programmable self-assembly, *Control Systems, IEEE* 27 (4) 43–56 (2007).
- [30] A. Murugan, J. Zou, M. P. Brenner, Undesired usage and the robust self-assembly of heterogeneous structures, *Nature Communications* 6 6203 (2015).
- [31] R. J. Siezen, M. R. Fisch, C. Slingsby, G. B. Benedek, Opacification of gamma-crystallin solutions from calf lens in relation to cold cataract formation, *PNAS* 82 (6) 1701–1705 (1985).
- [32] A. E. S. Van Driessche, N. Van Gerven, P. H. H. Bomans, R. R. M. Joosten, H. Friedrich, D. Gil-Carton, N. A. J. M. Sommerdijk, M. Sleutel, Molecular nucleation mechanisms and control strategies for crystal polymorph selection, *Nature* 556 89 (2018).
- [33] A. Ciesielski, P. J. Szabelski, W. Rżysko, A. Cadeddu, T. R. Cook, P. J. Stang, P. Samorì, Concentration-Dependent Supramolecular Engineering of Hydrogen-Bonded Nanostructures at Surfaces: Predicting Self-Assembly in 2D, *J. Am. Chem. Soc.* 135 (18) 6942–6950 (2013).
- [34] D. J. Ashton, N. B. Wilding, Three-body interactions in complex fluids: Virial coefficients from simulation finite-size effects, *J. Chem. Phys.* 140 (24) 244118 (2014).
- [35] S. C. Glotzer, M. J. Solomon, Anisotropy of building blocks and their assembly into complex structures, *Nature Materials* 6 557–562 (2007).
- [36] M. A. Bates, D. Frenkel, Influence of polydispersity on the phase behavior of colloidal liquid crystals: A Monte Carlo simulation study, *J. Chem. Phys.* 109 (14) 6193–6199 (1998).
- [37] J. A. Millan, D. Ortiz, S. C. Glotzer, Effect of shape on the self-assembly of faceted patchy nanoplates with irregular shape into tiling patterns, *Soft Matter* 11 1386–1396 (2015).
- [38] B. Senyuk, Q. Liu, S. He, R. D. Kamien, R. B. Kusner, T. C. Lubensky, I. I. Smalyukh, Topological colloids, *Nature* 493 200–205 (2013).

- [39] Q. Chen, S. C. Bae, S. Granick, Directed self-assembly of a colloidal kagome lattice, *Nature* 469 (7330) 381–384 (2011).
- [40] Q. Chen, E. Diesel, J. K. Whitmer, S. C. Bae, E. Luijten, S. Granick, Triblock colloids for directed self-assembly, *J. Am. Chem. Soc.* 133 (20) 7725–7727 (2011).
- [41] D. J. Kraft, W. S. Vlug, C. M. van Kats, A. van Blaaderen, A. Imhof, W. K. Kegel, Self-Assembly of Colloids with Liquid Protrusions, *J. Am. Chem. Soc.* 131 (3) 1182–1186 (2009).
- [42] D. J. Kraft, R. Ni, F. Smallenburg, M. Hermes, K. Yoon, D. A. Weitz, A. van Blaaderen, J. Groenewold, M. Dijkstra, W. K. Kegel, Surface roughness directed self-assembly of patchy particles into colloidal micelles, *PNAS* 109 (27) 10787–10792 (2012).
- [43] D. J. Ashton, S. J. Ivell, R. P. A. Dullens, R. L. Jack, N. B. Wilding, D. G. A. L. Aarts, Self-assembly and crystallisation of indented colloids at a planar wall, *Soft Matter* 11 6089–6098 (2015).
- [44] J. A. Champion, Y. K. Katare, S. Mitragotri, Particle shape: A new design parameter for micro- and nanoscale drug delivery carriers, *Journal of Controlled Release* 121 (1) 3–9 (2007).
- [45] M. Youssef, T. Hueckel, G.-R. Yi, S. Sacanna, Shape-shifting colloids via stimulated dewetting, *Nature Communications* 7 12216 (2016).
- [46] T. D. Nguyen, E. Jankowski, S. C. Glotzer, Self-Assembly and Reconfigurability of Shape-Shifting Particles, *ACS Nano* 5 (11) 8892–8903 (2011).
- [47] J.-W. Yoo, S. Mitragotri, Polymer particles that switch shape in response to a stimulus, *PNAS* 107 (25) 11205–11210 (2010).
- [48] É. Duguet, C. Hubert, C. Chomette, A. Perro, S. Ravaine, Patchy colloidal particles for programmed self-assembly, *Comptes Rendus Chimie*
- [49] F. Romano, F. Sciortino, Two dimensional assembly of triblock Janus particles into crystal phases in the two bond per patch limit, *Soft Matter* 7 5799–5804 (2011).
- [50] F. Smallenburg, F. Sciortino, Liquids more stable than crystals in particles with limited valence and flexible bonds, *Nature Physics* 9 554–558 (2013).
- [51] G. Doppelbauer, E. G. Noya, E. Bianchi, G. Kahl, Self-assembly scenarios of patchy colloidal particles, *Soft Matter* 8 7768–7772 (2012).
- [52] G. Doppelbauer, E. Bianchi, G. Kahl, Self-assembly scenarios of patchy colloidal particles in two dimensions, *Journal of Physics: Condensed Matter* 22 (10) 104105 (2010).
- [53] H. Shin, K. S. Schweizer, Theory of two-dimensional self-assembly of janus colloids: crystallization and orientational ordering, *Soft Matter* 10 262–274 (2014).

- [54] A. M. Jackson, J. W. Myerson, F. Stellacci, Spontaneous assembly of subnanometre-ordered domains in the ligand shell of monolayer-protected nanoparticles, *Nature Materials* 3 330–336 (2004).
- [55] I. C. Pons-Siepermann, S. C. Glotzer, Design of patchy particles using ternary self-assembled monolayers, *Soft Matter* 8 6226–6231 (2012).
- [56] I. C. Pons-Siepermann, S. C. Glotzer, Design of Patchy Particles Using Quaternary Self-Assembled Monolayers, *ACS Nano* 6 (5) 3919–3924 (2012).
- [57] F. Romano, F. Sciortino, Colloidal self-assembly: Patchy from the bottom up, *Nature Materials* 10 (3) 171–173 (2011).
- [58] S. Whitelam, Control of Pathways and Yields of Protein Crystallization through the Interplay of Nonspecific and Specific Attractions, *Phys. Rev. Lett.* 105 088102 (2010).
- [59] J. Madge, M. A. Miller, Design strategies for self-assembly of discrete targets, *J. Chem. Phys.* 143 (4) 044905 (2015).
- [60] W. B. Rogers, J. C. Crocker, Direct measurements of DNA-mediated colloidal interactions and their quantitative modeling, *PNAS* 108 (38) 15687–15692 (2011).
- [61] Y. Ke, L. L. Ong, W. M. Shih, P. Yin, Three-Dimensional Structures Self-Assembled from DNA Bricks, *Science* 338 (6111) 1177–1183 (2012).
- [62] A. Reinhardt, D. Frenkel, Numerical Evidence for Nucleated Self-Assembly of DNA Brick Structures, *Phys. Rev. Lett.* 112 238103 (2014).
- [63] W. M. Jacobs, A. Reinhardt, D. Frenkel, Rational design of self-assembly pathways for complex multicomponent structures, *PNAS* 112 (20) 6313–6318 (2015).
- [64] J. D. Halverson, A. V. Tkachenko, DNA-programmed mesoscopic architecture, *Phys. Rev. E* 87 062310 (2013).
- [65] Zhang, A. S. Keys, T. Chen, S. C. Glotzer, Self-Assembly of Patchy Particles into Diamond Structures through Molecular Mimicry, *Langmuir* 21 (25) 11547–11551 (2005).
- [66] G. Doppelbauer, E. G. Noya, E. Bianchi, G. Kahl, Competing ordered structures formed by particles with a regular tetrahedral patch decoration, *Journal of Physics: Condensed Matter* 24 (28) 284124 (2012).
- [67] S. Whitelam, R. L. Jack, The Statistical Mechanics of Dynamic Pathways to Self-Assembly, *Annual Review of Physical Chemistry* 66 (1) 143–163 (2015).
- [68] J. Grant, R. L. Jack, S. Whitelam, Analyzing mechanisms and microscopic reversibility of self-assembly, *J. Chem. Phys.* 135 (21) 214505 (2011).

- [69] G. M. Whitesides, M. Boncheva, Beyond molecules: Self-assembly of mesoscopic and macroscopic components, *PNAS* 99 (8) 4769–4774 (2002).
- [70] D. C. Rapaport, Role of Reversibility in Viral Capsid Growth: A Paradigm for Self-Assembly, *Phys. Rev. Lett.* 101 186101 (2008).
- [71] R. L. Jack, M. F. Hagan, D. Chandler, Fluctuation-dissipation ratios in the dynamics of self-assembly, *Phys. Rev. E* 76 021119 (2007).
- [72] I. Staneva, D. Frenkel, The role of non-specific interactions in a patchy model of protein crystallization, *J. Chem. Phys.* 143 (19) 194511 (2015).
- [73] M. G. Noro, D. Frenkel, Extended corresponding-states behavior for particles with variable range attractions, *J. Chem. Phys.* 113 2941–2944 (2000).
- [74] T. N. Carpency, J. D. Gunton, J. M. Rickman, Phase behavior of patchy spheroidal fluids, *J. Chem. Phys.* 145 (21) 214904 (2016).
- [75] T. K. Haxton, S. Whitelam, Design rules for the self-assembly of a protein crystal, *Soft Matter* 8 3558–3562 (2012).
- [76] C.-A. Palma, M. Cecchini, P. Samori, Predicting self-assembly: from empirism to determinism, *Chem. Soc. Rev.* 41 3713–3730 (2012).
- [77] Y. Yin, Y. Lu, B. Gates, Y. Xia, Template-Assisted Self-Assembly: A Practical Route to Complex Aggregates of Monodispersed Colloids with Well-Defined Sizes, Shapes, and Structures, *J. Am. Chem. Soc.* 123 (36) 8718–8729 (2001).
- [78] M. Bieri, M. Treier, J. Cai, K. Ait-Mansour, P. Ruffieux, O. Gröning, P. Gröning, M. Kastler, R. Rieger, X. Feng, K. Müllen, R. Fasel, Porous graphenes: two-dimensional polymer synthesis with atomic precision, *Chem. Commun.* 0 6919–6921 (2009).
- [79] M. F. Hagan, Controlling viral capsid assembly with templating, *Phys. Rev. E* 77 051904 (2008).
- [80] Y. Zhang, F. Lu, D. van der Lelie, O. Gang, Continuous Phase Transformation in Nanocube Assemblies, *Phys. Rev. Lett.* 107 135701 (2011).
- [81] C. J. Fullerton, R. L. Jack, Optimising self-assembly through time-dependent interactions, *J. Chem. Phys.* 145 (24) 244505 (2016).
- [82] B. Sweeney, T. Zhang, R. Schwartz, Exploring the parameter space of complex self-assembly through virus capsid models, *Biophysical Journal* 94 (3) 772–783 (2008).
- [83] M. B. Zanjani, I. C. Jenkins, J. C. Crocker, T. Sinno, Colloidal cluster assembly into ordered superstructures via engineered directional binding, *ACS Nano* 10 (12) 11280–11289 (2016).

- [84] J. Billeter, R. Pelcovits, Simulations of Liquid Crystals, *Computers in Physics* 12 (5) 440–448 (1998).
- [85] T. C. Moore, C. R. Iacovella, C. McCabe, Derivation of coarse-grained potentials via multistate iterative Boltzmann inversion, *J. Chem. Phys.* 140 (22) 224104 (2014).
- [86] N. B. Wilding, A nonequilibrium Monte Carlo approach to potential refinement in inverse problems, *J. Chem. Phys.* 119 (23) 12163–12168 (2003).
- [87] D. Klotsa, R. L. Jack, Predicting the self-assembly of a model colloidal crystal, *Soft Matter* 7 6294–6303 (2011).
- [88] D. Klotsa, R. L. Jack, Controlling crystal self-assembly using a real-time feedback scheme, *J. Chem. Phys.* 138 (9) 094502 (2013).
- [89] T. D. Nguyen, B. A. Schultz, N. A. Kotov, S. C. Glotzer, Generic, phenomenological, on-the-fly renormalized repulsion model for self-limited organization of terminal supraparticle assemblies, *PNAS* 112 (25) E3161–E3168 (2015).
- [90] T. K. Haxton, L. O. Hedges, S. Whitlam, Crystallization and arrest mechanisms of model colloids, *Soft Matter* 11 9307–9320 (2015).
- [91] A. W. Wilber, J. P. K. Doye, A. A. Louis, E. G. Noya, M. A. Miller, P. Wong, Reversible self-assembly of patchy particles into monodisperse icosahedral clusters, *J. Chem. Phys.* 127 (8) 085106 (2007).
- [92] C. L. Phillips, A learning heuristic for space mapping and searching self-organizing systems using adaptive mesh refinement, *Journal of Computational Physics* 272 799 – 813 (2014).
- [93] C. L. Phillips, G. A. Voth, Discovering crystals using shape matching and machine learning, *Soft Matter* 9 8552–8568 (2013).
- [94] J. Liu, E. Luijten, Rejection-Free Geometric Cluster Algorithm for Complex Fluids, *Phys. Rev. Lett.* 92 035504 (2004).
- [95] D. J. Ashton, J. Liu, E. Luijten, N. B. Wilding, Monte carlo cluster algorithm for fluid phase transitions in highly size-asymmetrical binary mixtures, *J. Chem. Phys.* 133 (19) 194102 (2010).
- [96] J. P. Nilmeier, G. E. Crooks, D. D. L. Minh, J. D. Chodera, Nonequilibrium candidate Monte Carlo is an efficient tool for equilibrium simulation, *PNAS* 108 (45) E1009–E1018 (2011).
- [97] N. B. Wilding, Computer simulation of fluid phase transitions, *Am. J. Phys.* 69 1147–1155 (2001).
- [98] D. Frenkel, B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications*, Computational Science Series, Elsevier Science, 2001.

- [99] M. Antlanger, G. Doppelbauer, G. Kahl, On the stability of Archimedean tilings formed by patchy particles, *J. Phys.: Condens. Matter* 23 (40) 404206 (2011).
- [100] E. Bianchi, G. Doppelbauer, L. Filion, M. Dijkstra, G. Kahl, Predicting patchy particle crystals: Variable box shape simulations and evolutionary algorithms, *J. Chem. Phys.* 136 (21) 214102 (2012).
- [101] S. Whitelam, P. L. Geissler, Avoiding unphysical kinetic traps in Monte Carlo simulations of strongly attractive particles, *J. Chem. Phys.* 127 (15) 154101 (2007).
- [102] R. J. Allen, C. Valeriani, P. R. ten Wolde, Forward flux sampling for rare event simulations, *Journal of Physics: Condensed Matter* 21 (46) 463102 (2009).
- [103] N. B. Becker, R. J. Allen, P. R. ten Wolde, Non-stationary forward flux sampling, *J. Chem. Phys.* 136 (17) 174118 (2012).
- [104] C. C. Battaile, The Kinetic Monte Carlo method: Foundation, implementation, and application, *Computer Methods in Applied Mechanics and Engineering* 197 (41) 3386–3398 (2008).
- [105] S. Dey, J. Saha, Solvent-free, molecular-level modeling of self-assembling amphiphiles in water, *Phys. Rev. E* 95 023315 (2017).
- [106] A. Jusufi, A. Z. Panagiotopoulos, Explicit- and Implicit-Solvent Simulations of Micellization in Surfactant Solutions, *Langmuir* 31 (11) 3283–3292 (2015).
- [107] M. E. J. Newman, G. T. Barkema, *Monte Carlo Methods in Statistical Physics*, Oxford University Press, 2009.
- [108] V. I. Manousiouthakis, M. W. Deem, Strict detailed balance is unnecessary in Monte Carlo simulation, *J. Chem. Phys.* 110 (6) 2753–2756 (1999).
- [109] G. Grimmett, *Probability and Random Processes*, 3rd Edition, Oxford University Press, Oxford, 2001.
- [110] N. van Kampen, *Stochastic Processes in Physics and Chemistry*, 3rd Edition, North-Holland Personal Library, Elsevier Science, 2007.
- [111] H. Behringer, R. Eichhorn, Brownian dynamics simulations with hard-body interactions: Spherical particles, *J. Chem. Phys.* 137 (16) 164108 (2012).
- [112] H. K. Shin, C. Kim, P. Talkner, E. K. Lee, Brownian motion from molecular dynamics, *Chemical Physics* 375 (2) 316 – 326 (2010).
- [113] K. Kikuchi, M. Yoshida, T. Maekawa, H. Watanabe, Metropolis Monte Carlo method as a numerical technique to solve the Fokker—Planck equation, *Chemical Physics Letters* 185 (3) 335–338 (1991).
- [114] G. Tiana, L. Sutto, R. Broglia, Use of the Metropolis algorithm to simulate the dynamics of protein chains, *Physica A: Statistical Mechanics and its Applications* 380 241–249 (2007).

- [115] E. Sanz, D. Marenduzzo, Dynamic Monte Carlo versus Brownian dynamics: A comparison for self-diffusion and crystallization in colloidal fluids, *J. Chem. Phys.* 132 (19).
- [116] A. Patti, A. Cuetos, Brownian dynamics and dynamic Monte Carlo simulations of isotropic and liquid crystal phases of anisotropic colloidal particles: A comparative study, *Phys. Rev. E* 86 011403 (2012).
- [117] A. Cuetos, A. Patti, Equivalence of Brownian dynamics and dynamic Monte Carlo simulations in multicomponent colloidal suspensions, *Phys. Rev. E* 92 022302 (2015).
- [118] S. Whitelam, Approximating the dynamical evolution of systems of strongly interacting overdamped particles, *Molecular Simulation* 37 (7) 606–612 (2011).
- [119] L. O. Hedges, Libvmmc. URL <http://vmmc.xyz>
- [120] N. Kern, D. Frenkel, Fluid–fluid coexistence in colloidal systems with short-ranged strongly directional attraction, *J. Chem. Phys.* 118 (21) 9882–9889 (2003).
- [121] T. Vissers, Z. Preisler, F. Smalenburg, M. Dijkstra, F. Sciortino, Predicting crystals of Janus colloids, *J. Chem. Phys.* 138 (16) 164505 (2013).
- [122] F. Romano, E. Sanz, P. Tartaglia, F. Sciortino, Phase diagram of trivalent and pentavalent patchy particles, *Journal of Physics: Condensed Matter* 24 (6) 064113 (2012).
- [123] A. Fortini, E. Sanz, M. Dijkstra, Crystallization and gelation in colloidal systems with short-ranged attractive interactions, *Phys. Rev. E* 78 041402 (2008).
- [124] R. Diestel, *Graph Theory*, 5th Edition, Graduate Texts in Mathematics, 173, Berlin, Heidelberg : Springer Berlin Heidelberg : Imprint: Springer, 2017.
- [125] L. Isella, Y. Drossinos, Langevin agglomeration of nanoparticles interacting via a central potential, *Phys. Rev. E* 82 011404 (2010).
- [126] S. Dempe, G. Eichfelder, J. Fliege, On the effects of combining objectives in multi-objective optimization, *J. Math Meth Oper Res* 82 (1) 1–18 (2015).
- [127] T. A. Witten, L. M. Sander, Diffusion-Limited Aggregation, a Kinetic Critical Phenomenon, *Phys. Rev. Lett.* 47 1400–1403 (1981).
- [128] C.-A. Palma, P. Samorì, M. Cecchini, Atomistic Simulations of 2D Bicomponent Self-Assembly: From Molecular Recognition to Self-Healing, *J. Am. Chem. Soc.* 132 (50) 17880–17885 (2010).
- [129] E. Talbi, *Metaheuristics: From Design to Implementation*, Wiley Series on Parallel and Distributed Computing, Wiley, 2009.

- [130] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, Optimization by Simulated Annealing, *Science* 220 (4598) 671–680 (1983).
- [131] F. Glover, M. Laguna, *Tabu Search*, Springer New York, New York, NY, 2013, pp. 3261–3362.
- [132] C. Blum, A. Roli, Metaheuristics in combinatorial optimization: Overview and conceptual comparison, *ACM Comput. Surv.* 35 (3) 268–308 (2003).
- [133] E. Marinari, G. Parisi, Simulated Tempering: A New Monte Carlo Scheme, *EPL (Europhysics Letters)* 19 (6) 451 (1992).
- [134] K. M. Malan, A. P. Engelbrecht, A survey of techniques for characterising fitness landscapes and some possible ways forward, *Inform. Sci.* 241 148–163 (2013).
- [135] G. Beveridge, R. Schechter, *Optimization: theory and practice*, International Student Edition Edition, McGraw-Hill Chemical Engineering Series, McGraw-Hill, 1970.
- [136] K. Sorensen, M. Sevaux, F. Glover, A History of Metaheuristics, [arXiv:1704.00853](https://arxiv.org/abs/1704.00853) (2017).
- [137] J. Blazewicz, K. Ecker, E. Pesch, G. Schmidt, J. Weglarz, *Scheduling Computer and Manufacturing Processes*, Springer-Verlag Berlin Heidelberg, 1996.
- [138] S. Rana, L. D. Whitley, R. Cogswell, Searching in the presence of noise, in: H.-M. Voigt, W. Ebeling, I. Rechenberg, H.-P. Schwefel (Eds.), *Parallel Problem Solving from Nature — PPSN IV: International Conference on Evolutionary Computation — The 4th International Conference on Parallel Problem Solving from Nature* Berlin, Germany, September 22–26, 1996 Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 1996, pp. 198–207.
- [139] M. B. Zanjani, J. C. Crocker, T. Sinno, Self-assembly with colloidal clusters: facile crystal design using connectivity landscape analysis, *Soft Matter* 13 (39) 7098–7105 (2017).
- [140] W. M. Jacobs, D. Frenkel, Self-assembly protocol design for periodic multicomponent structures, *Soft Matter* 11 8930–8938 (2015).
- [141] M. Basseur, A. Goeffon, Hill-climbing strategies on various landscapes: An empirical comparison, in: *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation, (GECCO '13)*, ACM, 2013, pp. 479–486.
- [142] R. G. William W. Cohen, D. Schuurmans, Probabilistic hill-climbing, in: S. Hanson, M. Kearns, T. Petsche, R. Rivest (Eds.), *Computational Learning Theory and Natural Learning Systems: Intersections Between Theory and Experiment*, Vol. 2, MIT Press, 1994, Ch. 11, pp. 171–181.
- [143] E. K. Burke, Y. Bykov, The late acceptance Hill-Climbing heuristic, *European Journal of Operational Research* 258 (1) 70–78 (2017).

- [144] B. Levitan, S. Kauffman, Adaptive walks with noisy fitness measurements, *Molecular Diversity* 1 (1) 53–68 (1995).
- [145] S. Nowak, J. Krug, Analysis of adaptive walks on NK fitness landscapes with different interaction schemes, *Journal of Statistical Mechanics: Theory and Experiment* 6 06014 (2015).
- [146] S. J. S. J. Russell, *Artificial Intelligence : A Modern Approach*, 3rd Edition, Prentice Hall series in artificial intelligence, Pearson, Boston, [Mass.] ; London, 2010.
- [147] M. Basseur, A. Goëffon, Climbing combinatorial fitness landscapes, *Applied Soft Computing* 30 688–704 (2015).
- [148] M. Kendall, A. Stuart, *The Advanced Theory of Statistics*, 4th Edition, Vol. 1, Griffin, London, 1977.
- [149] J. M. Fitzpatrick, J. J. Grefenstette, Genetic algorithms in noisy environments, *Machine Learning* 3 (2) 101–120 (1988).
- [150] Y. Jin, J. Branke, Evolutionary Optimization in Uncertain Environments-a Survey, *IEEE Transactions on Evolutionary Computation* 9 (3) 303–317 (2005).
- [151] F. Stonedahl, S. H. Stonedahl, Heuristics for sampling repetitions in noisy landscapes with fitness caching, in: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, GECCO '10, ACM, New York, NY, USA, 2010, pp. 273–280.
- [152] J. Liu, M. Fairbank, D. Pérez-Liébaná, S. M. Lucas, Optimal resampling for the noisy OneMax problem, [arXiv:1607.06641](#) (2016).
- [153] J. F. Kenney, E. S. Keeping, *Mathematics of Statistics*, 3rd Edition, Vol. 1, Van Nostrand, Princeton, N.J. ; London, 1954.
- [154] D. Whitley, S. Rana, J. Dzubera, K. E. Mathias, Evaluating evolutionary algorithms, *Artificial Intelligence* 85 (1) 245–276 (1996).
- [155] L. Di Gaspero, A. Schaerf, Multi-neighbourhood local search with application to course timetabling, in: E. Burke, P. De Causmaecker (Eds.), Practice and Theory of Automated Timetabling IV, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, pp. 262–275.