Unifying a Geometric Framework of Evolutionary Algorithms and Elementary Landscapes Theory

Submitted by Marcos Diez García to the University of Exeter
as a thesis for the degree of
Doctor of Philosophy in Computer Science,
January 2021.

Abstract

Evolutionary algorithms (EAs) are randomised general-purpose strategies, inspired by natural evolution, often used for finding (near) optimal solutions to problems in combinatorial optimisation. Over the last 50 years, many theoretical approaches in evolutionary computation have been developed to analyse the performance of EAs, design EAs or measure problem difficulty via fitness landscape analysis. An open challenge is to formally explain why a general class of EAs perform better, or worse, than others on a class of combinatorial problems across representations. However, the lack of a general unified theory of EAs and fitness landscapes, across problems and representations, makes it harder to characterise pairs of general classes of EAs and combinatorial problems where good performance can be guaranteed provably.

This thesis explores a unification between a geometric framework of EAs and elementary landscapes theory, not tied to a specific representation nor problem, with complementary strengths in the analysis of population-based EAs and combinatorial landscapes. This unification organises around three essential aspects: search space structure induced by crossovers, search behaviour of population-based EAs and structure of fitness landscapes. First, this thesis builds a crossover classification to systematically compare crossovers in the geometric framework and elementary landscapes theory, revealing a shared general subclass of crossovers: geometric recombination P-structures, which covers well-known crossovers. The crossover classification is then extended to a general framework for axiomatically analysing the population behaviour induced by crossover classes on associated EAs. This shows the shared general class of all EAs using geometric recombination P-structures, but no mutation, always do the same abstract form of convex evolutionary search. Finally, this thesis characterises a class of globally convex combinatorial landscapes shared by the geometric framework and elementary landscapes theory: abstract convex elementary landscapes. It is formally explained why geometric recombination P-structure EAs expectedly can outperform random search on abstract convex elementary landscapes related to low-order graph Laplacian eigenvalues. Altogether, this thesis paves a way towards a general unified theory of EAs and combinatorial fitness landscapes.

Acknowledgements

Completing this thesis is the most memorable experience I have had since January 2017, when I began my journey as a doctoral student at the University of Exeter. I would like to thank here all the great people who helped me to make it possible.

I am profoundly grateful to my supervisor Alberto Moraglio and co-supervisor Jonathan Fieldsend who guided me over the course of my doctoral study, providing me with most valuable feedback and fostering my growth as a doctoral researcher. Also, I would like to thank my advisors Ke Li and Lorenzo Livi for the helpful discussions to focus my research, as well as my pastoral tutor Chunbo Luo for encouraging me. I thank the University of Exeter for providing everything that I needed to complete my doctoral programme successfully.

From the evolutionary computation community I wish to thank all the researchers with whom I exchanged many ideas that stimulated and inspired my research such as Peter F. Stadler, Sébastien Verel, Francisco Chicano, Gabriela Ochoa, among many others. From outside the evolutionary computation community, I wish to thank two excellent mathematicians, Pierre Duchet and Manoj Changat, who kindly helped me solve a couple of doubts concerning abstract convexity theory.

Many thanks to my close fellow students Tina Malalanirainy, Antonis Loizou, Abdulaziz Almutairi, Joe Billingsley, Matthew McGuigan, Nick Sanders, Dongya Wang, Ning Wang, Gopika Rajan, Juan Zhang, and Cristina Alexandru for creating a welcoming and friendly atmosphere that made my journey more pleasant.

I would like to thank my dear family for their unconditional love and personal support to whom I owe a vital part of many aspects of my life. I thank Donald and Barbara Bligh for being my host family in Exeter over two years. Thanks to Chris Croft, my yoga instructor, as well as Tom Collingridge and Helen Kingdon, my Wing Chun martial art instructors, who contributed to my good mental and physical health.

I am very grateful to Franz Rothlauf and Achim D. Brucker for examining my thesis.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Author Declaration

This thesis is based on two co-authored publications I wrote during my doctoral research period at the University of Exeter. Both publications were co-authored with Alberto Moraglio whose role was supervising and providing feedback. The publications, and chapters that they have contributed to, are the following:

- The work 'Bridging Elementary Landscapes and a Geometric Theory of Evolutionary Algorithms: First Steps' in the proceedings of the 15th international conference 'Parallel Problem Solving from Nature – PPSN XV' [48] contributes to Chapter 5 and Chapter 11.
- The work 'A Unifying View on Recombination Spaces and Abstract Convex Evolutionary Search' in the proceedings of the 19th European conference 'Evolutionary Computation in Combinatorial Optimisation – EvoCOP 2019' [49] contributes to Chapter 5 and Chapter 8.

All illustrations or figures in this thesis are my own work.

Abbreviations

DE	differential evolution
DND	discrete nodal domain
EA	evolutionary algorithm
EC	evolutionary computation
EDA	estimation of distribution algorithm
EL	elementary landscape
ELT	elementary landscapes theory
EP	evolutionary programming
ES	evolution strategy
GA	genetic algorithm
GF	geometric framework
GP	genetic programming
HC	headless-chicken
NAE3SATP	not-all-equal 3-satisfiability problem
NFL	no-free-lunch
NP	non-deterministic polynomial time
P	polynomial time
PSO	particle swarm optimisation

TSP	travelling salesman problem
WPP	weight partitioning problem

Chapter 1

Introduction

A few years after the end of World War II, the transition to programmable digital computers and the opening of the field of evolutionary computation (EC) [4] gave birth to certain computer programs, called evolutionary algorithms (EAs), inspired by Darwin's theory of evolution by natural selection [27].

EAs are general-purpose, randomised, problem-solving strategies; that is, models of meta-heuristics [9, 52, 122]. EAs can accomplish a wide range of tasks in multiple domains. For instance, to search for solutions of mathematical optimisation problems, like the well known travelling salesman problem (TSP) [46]; or, to simulate the evolution of biological organisms, where the goal is not so much to optimise a fixed function as to adapt to a changing environment [32, 65, 77]. This thesis concerns EAs for mathematical optimisation problems, mainly those with a finite number of possible solutions (i.e. combinatorial optimisation problems). Conventional EAs consist of only a few key components reminiscent of natural evolution [4]:

- A *population of individuals* where each individual is a symbolic form (called 'genotype') that corresponds to a candidate solution (called 'phenotype') according to a genotype-phenotype function (called 'representation').
- A *fitness* function to evaluate the quality of candidate solutions, based on the objective function and possibly other information about the problem.
- A *selection* operator that selects the fittest individuals from a given population depending on their fitness function evaluation.
- A *crossover* and *mutation* operator that transform selected individuals into other individuals.

These components are enough for an EA to carry out an evolutionary search that, broadly speaking, consists in creating an initial population and then iteratively

transforming it to generate new (expectedly fitter) populations via selection, crossover, and mutation operations for as long as needed.

It is no surprise that EAs are often applied to find approximate solutions of optimisation problems since EAs strike a good balance between their general problem-solving capabilities and efficiency [4, 53, 96]. For example, consider the problem of finding shortest-paths between every pair of vertices in a directed weighted graph[1] with n vertices and m edges. This is the well known all-pairs shortest-path problem with countless applications in computer science. Doerr and others [37] show an EA that finds an optimal solution to the all-pairs shortest-path problem within $\mathcal{O}(n^3 \log n)$ fitness evaluations, compared with the computational time complexity $\mathcal{O}(n^2 \log n + nm)$ of best-specific solvers based on Johnson's algorithm [72].

Designing effective and efficient EAs is not simple though. A recommendation in EC [4, 96] is to design classes of EAs that target specific classes of problems by incorporating heuristic information of those problems into the EAs. This is in keeping with the no-free-lunch (NFL) theorems [68, 155], from which (loosely speaking) follows that any EA performs as 'good' as any other EA, or pure random search, if their performance is averaged across all problems with discrete objective functions. But this raises the more difficult issue of formally explaining why certain EA classes perform well (or poorly) on certain problem classes, which motivates the context of this thesis: general theories of EAs and fitness landscapes of combinatorial problems.

The notion of fitness landscapes in EC [120] is a valuable tool to deal with the aforesaid issue because it provides a means to identify and analyse what features of problems affect EA performance, which can be used to guide EA design. Fitness landscapes consist of a fitness function defined on the set of all candidate solutions for a given problem and a topological structure, normally associated with crossover or mutation, interrelating those solutions. Informally, a feature refers to a property of a fitness landscape that may be known a priori or not (e.g. whether the fitness function is linear or non-linear, or a particular distribution of local optima is present).

Currently, however, it is still unclear how the design and performance analysis of general EA classes, as well as the analysis of general fitness landscapes associated with problem classes, can be integrated within a coherent mathematical framework. In other words, it is unclear if a general unified theory of EAs and fitness landscapes across different problems and solution representations is possible. This is mainly due to a lack of common theoretical foundations in EC between research on EAs and fitness landscapes [4, 33, 97, 120, 141][95, chs. 2–3,7], despite the increasing efforts to consolidate both areas. This thesis will contribute towards such unification.

[1] Assuming there is at least one path between any two vertices and no negative weight cycles.

1.1 Aim and Motivations

The aim of this thesis is to explore if and how it is possible to formally unify the foundations of a general theory of EAs and a general theory of combinatorial fitness landscapes, known in EC respectively as: the geometric framework (GF) of EAs proposed by Moraglio [100], and elementary landscapes theory (ELT) proposed by Stadler [136]. To motivate and clarify what this thesis means by their unification, let us overview first some of the most salient aspects of the GF and ELT.

1.1.1 Overview of the Geometric Framework of EAs

The GF is a theory about a class of generalised EAs that comprises many pre-existing EAs, based on a geometric generalisation of traditional mutation and crossover operators across problems and solution representations [100]. Among them, a prominent class are geometric-crossover EAs (without mutation). Geometric crossovers are generalised crossovers associated with a generic notion of metric distance [34, 100], which can be designed in a principled manner by specifying a distance and comprise many crossovers used in practice for many solution representations (e.g. trees, sequences, vectors of reals, or permutations). Some applications of geometric crossovers are [100]: filtering redundancies in genotype-phenotype maps to prevent loss of search performance, removing undesired biases inherent to evolutionary search, or generalising other meta-heuristics like particle swarm optimisation [52, 80, 105]. Provably, all geometric-crossover EAs do the same form of search regardless of a specific problem and solution representation: abstract convex evolutionary search [100]. Also, a general runtime analysis in the GF [104] shows that certain geometric-crossover EAs can exponentially outperform pure random search on a certain class of abstract convex fitness landscapes [101] across problems and solution representations.

1.1.2 Overview of Elementary Landscapes Theory

ELT is a theory built upon a general class of combinatorial fitness landscapes, called elementary landscapes, whose underlying structure of local optima tends to respect certain symmetries and correlation among fitness values [35, 136, 137]. Elementary landscapes appear for instance in physics models of real phenomena like spin-glass, software engineering problems and various combinatorial optimisation problems in the P or NP-complete computational complexity classes (e.g. linear assignment, TSP, or graph colouring) [85, 91, 121]. Moreover, elementary landscapes correspond to eigenfunctions of certain discrete Laplacian operators, in turn,

associated with mutation operators or generalised crossovers called recombination P-structures [51, 139, 150]. This makes the powerful tools of spectral graph theory [13] available in ELT, thus opening the door to a large number of possible applications of Laplacian eigenfunctions [5, 7, 8, 20, 56, 84, 87, 136, 140]: general methodologies for analysis of elementary landscapes, guiding local search algorithms, analysis of (recombination P-structure) random walks, among many others.

1.1.3 Towards a Unified Theory of EAs and Landscapes

By unifying the GF and ELT, this thesis means to develop a theory about a general class of EAs associated with a general class of fitness landscapes both shared by the GF and ELT. That is, to take a step towards a general theory of EAs and fitness landscapes, across problems and solution representations, where it is possible to:

- Design useful crossovers, whether based on geometric crossover or recombination P-structures, in a principled manner.
- Formalise the search behaviour of EAs based on geometric crossovers or recombination P-structures, possibly as some generalised form of abstract convex evolutionary search.
- Identify and analyse how features of elementary landscapes or abstract convex landscapes, associated with relevant known problems, impact the performance of EAs based on geometric crossovers or recombination P-structures.

The principal reason motivating this thesis to focus on the three previous aspects is that they correspond to three key areas where, in view of Sections 1.1.1 and 1.1.2, the GF and ELT can contribute and complement each other towards one such general theory of EAs and fitness landscapes. That is, crossover operators, evolutionary search and fitness landscapes, as follows:

- *Crossovers.* Geometric crossovers occur often in practice and can be designed in a principled manner across different representations [100]; however, geometric crossovers are inherently limited to metric spaces and thus exclude useful non-geometric crossovers [103] such as Koza's subtree swap [86] or Davis's order [31]. By contrast, few crossover examples of recombination P-structures are known [51, 150], basically multi-point string-based crossovers (e.g. one-point or uniform), and recombination P-structures have no established design principles. Nevertheless, recombination P-structures do not require metrics and thus do not exclude non-geometric crossovers in principle.

- *Evolutionary search.* The search carried by geometric-crossover EAs can be formalised across problems and representations, and it is always abstract convex evolutionary search [100]; however, abstract convex evolutionary search is not guaranteed in the presence of mutation or non-geometric crossovers. By contrast, ELT lacks a model of population-based EAs for recombination P-structures [85]; instead, evolutionary search is modelled as recombination P-structure random walks [140, 150]. Nevertheless, recombination P-structure random walks inherently do a form of macro-mutation (or 'headless-chicken' crossover), similar to that shown independently by Jones [73], and are not necessarily limited to metric spaces.

- *Fitness landscapes.* For certain classes of abstract convex fitness landscapes, certain geometric-crossover EAs can produce offspring populations whose fitness is on average not worse than that of their parent populations [101], or expectedly achieve polynomial runtime performance [104]. However, it is unclear what kind of optimisation problems characterise such abstract convex landscape classes, which is further obscured by the lack of landscape analysis tools in the GF to determine when a given problem corresponds to any of those classes [104]. By contrast, no performance analysis of any class of population-based EAs using recombination P-structures has been carried out on classes of elementary landscapes, except for recombination P-structure random walks [140]. Nevertheless, elementary landscapes are associated with several important problems in combinatorial optimisation and real-world scenarios, whose landscape features can be analysed with a wide range of tools based on spectral graph theory [85, 136, 139].

Thus unifying the GF and ELT in the above three areas will mean hereinafter that this thesis seeks to characterise (according to mathematical axioms of the GF and ELT) what classes of crossovers, classes of evolutionary search and classes of fitness landscapes do the GF and ELT have in common or not. As such, the methodology adopted in this thesis fundamentally consists in developing formal classifications to rigorously define, analyse and compare what those classes are.

Although this thesis discusses mutation operators in the GF and ELT, mutation operators are not the main focus of this thesis because the fundamental differences between GF and ELT stem from how they formalise crossover, namely geometric crossovers [100] and recombination P-structures [139], where a unification is precisely most needed.

Would such a general unified theory be possible at all? Whether the answer is

positive or not, exploring it is worthwhile either way. In the worst-case scenario, it would clarify the limits and potential practical implications of a general theory in EC built upon the GF and ELT. In the best-case scenario, it would reveal a profound and rigorous understanding of the inherent relationships between important classes of EAs and fitness landscapes as conceived in GF and ELT.

1.2 General Research Questions

Considering the aim and motivations mentioned in Section 1.1, this thesis investigates the following general research questions:

1. How can geometric crossovers, proposed in the GF, and recombination P-structures, proposed in ELT, be unified?
2. How can geometric-crossover EAs, proposed in the GF, and recombination P-structure random walks, proposed in the ELT, be unified?
3. How can abstract convex fitness landscapes, proposed in the GF, and elementary landscapes, proposed in the ELT, be unified?
4. What are the consequences of each of the previous unifications and how do they contribute to unify the GF and ELT under a common mathematical framework?

1.3 Contributions Overview

The overall contribution of this thesis is showing a three-fold integrated approach to unify the theoretical foundations of a geometric framework of EAs and elementary landscapes as described in Section 1.1, corresponding to three major contributions:

1. A *classification of crossovers* that simplifies and systematises the task of formally comparing new or existing crossovers with respect to classes of generalised crossovers in the GF and ELT, namely geometric crossovers and recombination P-structures, across problems and solution representations. This classification reveals geometric crossovers and recombination P-structures share a common general subclass of crossovers, which can be designed in a principled manner through a notion of distance associated with the problem at hand.
2. A *qualitative framework for abstract interval convex search of evolutionary algorithms*, built upon the classification of crossovers and abstract convexity

theory, that simplifies and systematises the task of comparing how different crossovers affect the search behaviour of EAs using them. This reveals the GF and ELT share a general class of EAs (without mutation), called geometric recombination P-structure EAs, which based on a crossover subclass shared between geometric crossovers and recombination P-structures. The population behaviour of geometric recombination P-structure EAs is described as a specific case of a general form of search which is the same across problems and representations: abstract interval convex evolutionary search.

3. *Abstract convex elementary landscapes* as a class of combinatorial fitness landscapes shared by the GF and ELT, which respect certain desirable conditions of symmetry, global convexity and correlation in fitness of candidate solutions, independent of a specific solution representation or problem. Among the examples of such landscapes are one-max fitness landscapes (with or without bounded perturbations) defined on binary sequences. This thesis reveals ELT provides tools to identify analytically certain kinds of global convexity present in abstract convex elementary landscapes in a direct manner. Also, this thesis justifies why certain EAs based on geometric crossover or recombination P-structures would expectedly outperform pure random search on abstract convex elementary landscapes.

Another contribution of this thesis is to provide an updated, comprehensive yet concise revision of the foundations of the GF and ELT most relevant to this thesis, including additional clarifications and examples. Besides making this thesis self-contained, this revision is a helpful resource for those who are not familiar with GF or ELT and a quick reminder of the essential ideas for those who are familiar already.

All formal results of this thesis are stated as lemmas, theorems or corollaries with a corresponding proof next to them. Any result taken from the literature is stated as a proposition, omitting its proof, with a bibliographical reference.

1.4 Organisation

This thesis has the following organisation. After this introduction, Chapter 2 places this thesis in a wider context of related research in EC by presenting a literature review. Then, the main body of this thesis is divided into Parts I, II and III corresponding to the three major contributions overviewed in Section 1.3: a classification of crossovers (Chapter 5), a qualitative framework of abstract interval

convex evolutionary search (Chapter 8) and abstract convex elementary landscapes (Chapter 11). In turn, each major contribution develops upon two corresponding preliminary chapters that revise specific background of the GF and ELT respectively: search spaces associated with mutation and crossover operators (Chapters 3 and 4), evolutionary search (Chapters 6 and 7) and fitness landscapes based on those mutation and crossover operators (Chapters 9 and 10). Chapter 12 concludes this thesis with a discussion of all contributions, limitations, and directions for future research. Figure 1.1 below illustrates the overall organisation of this thesis.

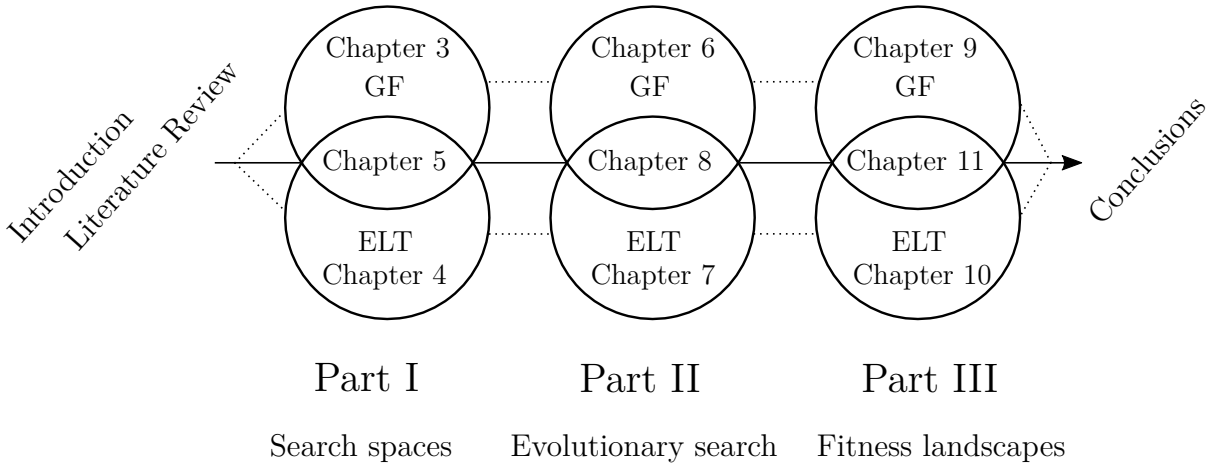


Figure 1.1. Thesis organisation.

Chapter 2

Literature Review

Evolutionary algorithms (EAs) and fitness landscapes constitute a fundamental part of EC since its origins [4]. Because both are generally difficult to analyse, much effort has been necessary to form a precise and coherent understanding about EAs and fitness landscapes [33, 42, 70, 97, 141][95, chs. 2–3,7] for a lack of solid theoretical foundations in EC. Still, it is far from clear how the existing theories of EAs [36] and fitness landscapes [120] make a general unified theory of EAs and fitness landscapes. Indeed, there is little indication if and how that unification is possible at all, except for a geometric framework (GF) of EAs [100]. If such a unified theory is not possible, then it is questionable if one coherent mathematical framework can rigorously explain why a certain class of EAs performs well (or poorly) on a certain class of fitness landscapes across different problems and solution representations. While this thesis does not aim to analyse performance of EA classes on fitness landscapes, the previous question adds to the motivations for exploring a theoretical unification between the GF and elementary landscapes theory (ELT) as introduced in Chapter 1.

2.1 Aim

This chapter reviews the literature in EC about other major theories or frameworks of EAs and fitness landscapes, rather than GF and ELT. The reason being Part I, Part II and Part III are self-contained and will cover all details or background of the GF and ELT to understand the contributions of this thesis outlined in Section 1.3. The aim of this literature review is two-fold. First, to provide additional context in EC where GF and ELT can be placed. Second, to clarify whether the theories and frameworks reviewed meet the requirements that motivated the GF [100] towards a general theory of EC:

(a) Explain rigorously why, when and how a certain class of EAs performs well (or poorly) on a certain class of problems.

(b) Identify and analyse how features of a fitness landscape class, associated with relevant known problems, affect the performance of a corresponding EA class.

(c) Inform about convergence and runtime of EAs.

(d) Guide the design of EAs, including mutation and crossover operators or parameters thereof, for a class of problems.

(e) Be independent from a specific problem and representation of solutions.

These requirements inform the topics of the literature review organised as in Section 2.2. Although the scope of this literature review is necessarily broad, it does not intend to be fully detailed nor exhaustive with regards to any of the items (a) to (e).

2.2 Organisation

This chapter has the following organisation. Section 2.3 reviews major families of EAs to provide preliminary context for subsequent sections. Section 2.4 reviews key issues and approaches involved in the design of EAs based on knowledge specific to problem classes. Section 2.5 reviews major theories about the performance of EAs, including population behaviour, convergence (Section 2.5.1) and runtime analysis (Section 2.5.2). Section 2.6 reviews the foundations of fitness landscapes and common approaches to analyse fitness landscapes (Section 2.6.1) addressing the question: what makes an EA perform well (or poorly) on a given optimisation problem? Finally, Section 2.7 contrasts various issues in the literature reviewed with a unification between the GF and ELT explored in this thesis as well as specific research questions to be addressed.

2.3 Major Families of Evolutionary Algorithms

The roots of EC trace back to three independent families of EAs around the mid 1960s: evolutionary programming (EP) due to Fogel, Owens and Walsh [45]; genetic algorithms (GAs) proposed by Holland [65]; and, evolution strategies (ESs) thanks to Rechenberg [118] and Schwefel [130]. Overviews of their seminal works and related historical remarks in EC are referred to in [4, 52, 96] and [33, ch. 2].

Fogel and his co-workers [45] proposed EP to model how natural evolution [27] creates intelligent beings and their ability to predict future events. EP simulates the evolution of a population of finite state machines (undergoing mutations, e.g. adding transition rules or states) according to how well they predict unobserved symbols from a given sequence of symbols (the 'environment'). Holland [65] conceived GAs to simulate how complex systems adapt to changing environments, as observed in global economies or immune systems for instance. To do so, individuals are described as symbolic strings reminiscent of genetic code, delegating to a genotype-phenotype function the issue of how those symbols represent a real system. Then, GAs simulate that system by successively generating populations subject to: a selection that favours individuals of above-average fitness (i.e. survival of fit or competitive individuals), and genetic variation due to crossover operations or (to a lesser extent) mutations. By contrast, Rechenberg [118] and Schwefel [130] addressed numerical optimisation problems in engineering with ESs. In ESs, individuals are directly encoded as the candidate solutions, these being real-valued vectors, and populations are as small as a single individual subject to mutation operations but no crossover.

The previous three families laid a basis for new EAs, whose differences often blurred. Koza proposed genetic programming (GP) [86] which, in plain words, teaches a computer how to automatically evolve 'intelligent' computer programs. GP differs from traditional EP, GAs and ESs in the representation of individuals as hierarchical tree-like structures of variable size, these representing programs' syntax trees. Two other types of EAs are particle swarm optimisation (PSO) [3, ch. 7][28, 80] and differential evolution (DE) [28, 142]. Both target global optimisation of possibly non-linear, non-differentiable and multi-modal functions. PSO and DE represent individuals as real vectors like ESs but use crossover and have different mutation operators. Another form of EAs that more recently gained attention in EC are estimation of distribution algorithms (EDAs) [36, ch. 9][106]. EDAs differ significantly from all aforesaid EAs: they do not evolve individuals in a population but evolve a probability distribution model (defined on a search space of candidate solutions). At each generation, the EDA samples promising candidate solutions from its current probabilistic model to generate an improved version of the model where sampling high-quality solutions is more likely. EDAs use neither genetic mutation nor crossover operators.

Many taxonomies [9, 122][36, ch. 9][52, ch. 17] helped to organise the vast collection of existing EAs, clarifying differences between EAs, suggesting new EA designs and correcting inaccurate terminology. But these classifications basically depend on

historical, subjective or superficial aspects of EAs. By contrast, formal hierarchies of EAs based on their search operators, behaviour or computational complexity are means to understand why certain EAs provably perform well or not on certain problems. However, rigorous taxonomies or hierarchy results linking EA classes with runtime performance are few and scattered [39, 63, 116], mostly for simple EAs and toy problems. Jones [73] classified various local search algorithms but less formally. Clearly, EC lags behind classic theory of computation [76, 88] in the sense that mathematical hierarchies of algorithms or Turing machines associated with problem complexity classes are well established since the 1960s.

2.4 Design of Evolutionary Algorithms

The design of conventional EAs for optimisation problems involves making choices or assumptions about various EA components, primarily the search operators, such as mutation or crossover, and the associated representation of candidate solutions.

One design approach is to represent candidate solutions indirectly by defining a genotype-phenotype function that maps a set of symbolic forms (i.e. genotypes) to a set of candidate solutions (i.e. phenotypes) [123]. For instance, in traditional GAs [53, 65], genotypes are fixed-length binary sequences, and standard genetic operators are single-bit flip mutation or one-point crossover. With indirect representations, familiar genetic operators can be chosen once and then reused for different problems by modifying the genotype-phenotype map. Rothlauf [123, 124] discusses in-depth other good reasons to use indirect representations over direct representations[1], which define search operators on phenotypes themselves, for instance: to represent complex real-world objects that are not easily manipulated by search operators directly, or to encode problem constraints that reduce the search space dimensionality as shown also by Davis [31].

However, Koza [86], Radcliffe and Surry [117, 143] argue that indirect representations are unnecessary and prevent search operators from fully exploiting problem-specific knowledge compared with search operators defined on phenotypes directly. Whether direct or indirect representations are used, many agree [52, 68, 96, 153] that EAs should be designed for a certain class of problems, to abandon the pure 'black box' setting where EAs make no assumptions about (thus not exploiting) the underlying structure of the problem at hand, in view of Wolpert and Macready's NFL theorems [155]. The NFL theorems are 'zero-sum' statements that,

[1]Effectively, direct representations equate to bijective genotype-phenotype functions.

in rough terms, imply the average performance of any two algorithms (including EAs and random search) is equal across all fixed objective functions with discrete domain and co-domain. See [93] for an updated complete review of NFL theorems.

Genetic operators and representations should be designed according to problem characteristics favourable to efficient evolutionary search, but how? Some theory-driven frameworks propose to do so via metric distances [38, 123, 124, 144], certain equivalence classes called *formae* [117, 143], crossover invariant subsets [98, 99] or permutation groups [125, 126] relating the search spaces and genetic operators. Overall, these frameworks take advantage of problems with certain correlation between candidate solutions, so similar solutions have similar fitness, and problem decomposability, so (near) optimal solutions can be built incrementally by genetic variation operators that combine and propagate the genetic code of promising candidate solutions through generations as in Goldberg's 'building blocks' hypothesis [53]. At the root of this hypothesis and the aforesaid frameworks is schema theory [53, 65] (see next Section 2.5).

One factor to consider when designing genetic operators is choosing parameter settings that favour good performance. The crossover bias parameter, for instance, controls how the genetic code of an offspring is built from a mixture of the genetic code of its parents. Interestingly, Chicano, Whitley and Alba [23] show closed-form expressions, computable in polynomial time, indicating which bias of uniform crossover maximises the expected fitness of the offspring, for k-bounded pseudo-Boolean optimisation problems [11] including the maximum k-satisfiability NP-hard problem [76] with at most $k > 0$ literals per clause. Their theoretical framework relies on decomposing the fitness function with respect to the Walsh basis [54, 151].

In practice though, it is still time-consuming and difficult to determine what EA components (i.e. representation, crossover, mutation, selection and replacement mechanisms, population size, etc.) are the best design choice for solving a certain problem, when there are plenty of EAs to choose from (Section 2.3). This is indeed part of the algorithm selection problem formulated by Rice [119] already in 1976, which, in plain words, consists in selecting the best algorithm from a given set of algorithms to solve a given problem. Since Rice's seminal work, a whole area of research has developed state-of-the-art automated systems [81] for the selection and configuration of EAs (and other meta-heuristics), during or prior to execution, based on empirical data collected from the specific problem to be solved.

Adopting the theoretical or empirical approaches to design efficient EAs comes with its own challenges [97]: the former may lack practicality and the latter rigour. Currently, research areas such as dynamic parameter control [36, ch. 6] and parametrised

computational complexity [36, ch. 4] are bridging theory and practice with mathematically rigorous but practice-oriented frameworks to design efficient EAs.

2.5 Performance of Evolutionary Algorithms

Schema theory, proposed in 1975 by Holland [65] and popularised by Goldberg [53], is the earliest well-known attempt to explain the population behaviour of a simple GA. Despite being regarded as the fundamental theory of GAs [53], schema theory, and its schema theorem in particular, has attracted considerable criticism [4, ch. B2.5][2, 125, 154]. One of the most recurrent reasons is that the schema theorem only gives a lower-bound on the expected fraction of a population in a given schema[2] after one generation, thus it is inexact even for one generation and proves nothing about the population behaviour over multiple generations.

Nevertheless, much of the criticism on Holland's schema theory was addressed in follow-up studies between 1991 and 2013 [2, 98, 99, 115–117, 125, 141, 148] showing how schemas are generalised beyond binary strings and traditional GAs to GP and other EAs (not based on binary string representations exclusively) and how certain generalisations of Holland's schema theorem are exact (passing from a lower-bound to an equality) as well as applicable over multiple generations.

Apart from schema theory, other theories inspired in statistical mechanics or quantitative genetics [42, 138] contributed to explain the population behaviour of EAs, even though they have received less attention compared with more recent advancements in EC theory [3, 36].

Holland [65] did not conceive GAs and schema theory particularly for solving optimisation problems [32], yet when GAs or other EAs are used for such purposes it is worth knowing if the EA will be able to find optimal solutions for the problem at hand and how long will it take to find them. This concerns two other aspects of EA performance besides population behaviour, namely convergence and runtime analysis [3, 70] reviewed next.

2.5.1 Convergence of Evolutionary Algorithms

Markov chain theory [19, 110] entered EC in the early 1990s, laying the foundations for the first rigorous analyses on the global convergence [4, ch. B2.3][95, ch. 8] of simple GAs. A notion of global convergence is that the probability of containing a global optimum in the current population tends to one as the number of generations

[2]Schemas describe subsets of n-dimensional binary strings sharing up to n bits at certain positions.

elapsed tends to infinity. Using this notion Rudolph [127] showed, for all non-constant bounded pseudo-Boolean functions [11] on fixed-length binary strings, that GAs using fitness-proportionate selection, any string-based crossover, and single-bit flip mutation, do not converge to any global optima unless an elitist version of selection is used (i.e. one where populations always keep the fittest individual[s] found so far). In a follow-up work, Rudolph [128] generalised that result to other EAs and arbitrary search spaces. A different approach is Vose's infinite-population model [149] of simple GAs: populations are described by vectors of proportions of individuals in the population, and convergence of population sequences is understood as trajectories towards fixed-points (i.e. attractors) in a dynamical system. But, as its name suggests, this infinite-population model becomes accurate in predicting such trajectories when the population size is large tending to infinity.

Convergence guarantees, however, are of little use per se because they do not respond to a more practical yet difficult question: how fast does an EA converge per iteration to an optimal set of solutions of a given problem? To answer this question usually involves a Markov chain analysis of the convergence rates of the EA, which measure a form of progress towards optima, as shown for instance by Schmitt and Rothlauf [129] for a simple GA or by Jun He and Lin [60] more recently for general finite-population EAs. But this subject is comparably less studied than EA runtime analysis (Section 2.5.2) because certain difficulties inherent to Markov chain analysis [3, ch. 2][4, chs. B2.2–B2.4][70, ch. 3] complicate answering the aforesaid question precisely, except for very simple EAs like the $(1+1)$ EA substantially restricting the representation of solutions and search operators. The typical $(1+1)$ EA uses single-individual populations, elitist selection, single-bit flip mutation and no crossover: in essence, a local hill-climb search algorithm [9, 52] for bit-string search spaces.

2.5.2 Runtime of Evolutionary Algorithms

Runtime analysis is the standard way to formally analyse the performance of EAs and other randomised meta-heuristics [36]. Essentially, runtime analysis concerns a simple question: how much time does a given EA take to find an optimal solution to a given problem? In relation to this question, this section briefly reviews what is meant by performance measure as well as the intuitive idea behind two related general approaches to runtime analysis.

Performance Measures

A usual performance measure is the expected number of fitness function evaluations elapsed until the EA finds an optimal solution (called expected optimisation time or expected runtime) [3, 36, 70]. Three reasons for using this measure are:

- It is independent of implementation details, from the specification of the EA itself to hardware details such as processor architecture or clock ticks, which otherwise would make a formal analysis and comparison of EA performances not viable or misleading. For example, due to the randomised nature of EAs, there may be high variance in the number of clock ticks between different runs of the same EA with the same configuration and input for the same problem.
- The number of fitness function evaluations takes into account population size, whereas a performance measure defined by the number of generations elapsed does not and thus ignores prohibitively large population sizes.
- It approximates well the performance observed in practice because fitness function evaluations are usually the most costly operation per generation.

However, according to Jansen and Zarges [71]: the expected runtime is not a practical performance measure because it assumes that EAs can recognise when they have found an individual that is an optimal solution, but optima of fitness functions are usually unknown and difficult to recognise. They propose performance measures based on fixed-budget analysis [71][36, ch. 5], an alternative to runtime analysis, which analyses the expected fitness function value obtained after a predetermined fixed number of function evaluations (the 'budget'). This and other practice-oriented performance measures [36, ch. 3] are not widely adopted yet, and few fixed-budget results are known for combinatorial optimisation problems like TSP [108].

Drift Analysis and Level-based Analysis

Runtime analysis has advanced significantly since the early pioneering work by Droste, Jansen and Wegener [39] on the very simple $(1 + 1)$ EA for simple pseudo-Boolean fitness functions [11] such as one-max (counting the number of `1` bits for an input bit-string). Thanks to two general methods, called drift analysis and level-based analysis, it is now possible to analyse the runtime of many EA variants (see Section 2.3) with or without populations, crossover, mutation, elitist or non-elitist selection, for toy problems and relevant combinatorial optimisation problems [3, 36, 63, 70, 99].

Drift analysis was Hajek's proposal in 1982 [57], later introduced in EC by He and Yao in 2001 [61] and more recently generalised by Lehre and Witt [89] under one unifying drift theorem that subsumes nearly all previous drift theorems. Overviews of drift analysis and related techniques are found in [3, 36, 70]. Drift theorems allow to derive bounds on the expected runtime of EAs by deriving bounds for the expected progress (the 'drift') between one population generation and the next, which is easier to analyse compared with the whole run, without having to directly analyse a Markov chain model of the EA (as in Section 2.5.1). This is possible by mapping via a drift function, also known as potential or distance function, each Markov state to a non-negative real number that measures the distance to the optima. Effectively, the drift function replaces a Markov chain model of the EA, where random variables describe populations, by a comparably easier one-dimensional random process, where random variables describe distances [3, ch. 2][61].

However, finding a suitable drift function is often difficult because it requires precise mathematical inspection of both the problem and EA on a case-by-case basis, particularly when dealing with population-based EAs such as GAs. This issue was addressed by Corus, Dang, Eremeev and Lehre [24] who proposed a level-based theorem, derived from a drift theorem and tailored to population-based EAs, which encapsulates a general drift function in a way that the user does not need to specify one. That is, no drift function is required when applying the level-based theorem to obtain bounds on the expected runtime of population-based EAs. But the level-based theorem still has certain prerequisites [24] that the user needs to deal with separately for each EA-problem pair, such as finding an appropriate partitioning of the search space via fitness level sets (hence called level-based analysis).

2.6 Fitness Landscapes

EC inherits the notion of fitness landscape [120] from what in evolutionary biology is originally called 'surface of selective values' as introduced by Wright [156] in 1932. His idea was to describe the complex dynamics of evolutionary processes with a visually appealing graphic, which metaphorically evokes physical landscapes with peaks and valleys seen in nature. But, in Wright's own words the fitness landscape metaphor is 'useless for mathematical purposes' [157] or, worse, can be misleading according to Jones [73].

Nevertheless, building upon early work of Kauffman [77] and Weinberger [152] about fitness landscapes, or energy landscapes as known in statistical physics [83], the current definition of fitness landscape in EC [120] was made precise by Jones [73]

and Stadler [136] in line with using EAs for mathematical optimisation. Briefly, a fitness landscape is a mathematical relation or tuple consisting of: a set of candidate solutions for a given problem, a topological structure (associated with mutation and crossover) imposed on that solution set, and a fitness function defined on the solution set. The former two define the search space structure of a fitness landscape. Thus fitness landscapes are fundamental to understand how EAs and optimisation problems relate with each other, and analysing this relationship is essentially the purpose fitness landscape analysis (see Section 2.6.1 below).

Note, however, that it may not be always practical or possible to define fitness landscapes mathematically in closed-form in the above sense. For instance, fitness values may be accessible not from a fitness function but from the output of a computer simulation whose internal details are unknown, or ignored, acting as a 'black box' for the EA [81]. The rest of this section mostly concerns (combinatorial) optimisation problems for which the fitness function is mathematically defined in closed-form but not everything about it is known necessarily (e.g. location of global optima), that is 'grey box' problems [153].

2.6.1 Fitness Landscape Analysis

What makes an EA perform well or poorly on a given optimisation problem? Fitness landscape analysis [82, 120] addresses this question from the problem's view, by identifying and analysing properties of the fitness function or its associated search space structure that affect EA performance, rather than the EA's algorithmic view of Section 2.5. This section overviews two common approaches to fitness landscape analysis.

Characterising Fitness Landscapes

The first approach consists in characterising classes of fitness landscapes that share a common property related to EA performance, so each class includes only those fitness landscapes of similar 'difficulty' with respect to such common property [69, 114]. A fitness landscape may be called 'difficult', for instance, if an EA searching in that landscape needs exponential expected runtime to find a (near) optimal solution; if the expected runtime is polynomial, it may be called 'easy'. Note problem difficulty here differs from the classical computational complexity of problems [76, 88] which is independent from the EA used and its expected runtime [36, 70].

The class of fitness landscapes formed by all linear, non-negative, pseudo-Boolean functions [11] on n-dimensional binary sequences is an example that is provably

easy for the simple $(1+1)$ EA since the expected runtime is $\mathcal{O}(n \log n)$ according to Droste, Jansen and Wegener [39]. Here the search space structure is that of binary Hamming graphs [14, 59] because the $(1+1)$ EA uses single-bit flip mutation as the search operator (i.e. adjacent vertices are one-flip away from each other).

Apart from linearity, Rothlauf [123, 124] shows that fitness landscapes with high locality can benefit the performance of EAs which exploit such property. Informally, when fitness landscapes have high locality, similar candidate solutions tend to have similar fitness; similarity may be defined in terms of Lipschitz continuity [12, 132], provided the search space and search operators are associated with a natural notion of metric distance (e.g. Hamming distance for Hamming graphs) [34]. According to McDermott [93], meta-heuristics whose search operators exploit locality avoid NFL theorems [68, 155] and thus outperform random search.

Smoothness (or ruggedness) is also a property of fitness landscapes classes where similar solutions tend (not) to have similar fitness. EC [120] normally formalises smoothness not in terms of Lipschitz continuity or distances but statistical correlation between fitness values. This, in fact, has been the tradition since the 1990s when Weinberger [152] and Hordijk [66] used time series auto-regressive models [19] to analyse the class of tunably-rugged NK-landscapes introduced by Kauffman [77, 78]. However, not all fitness landscapes with high correlation or smoothness lead to good EA performance. For example, the needle-in-a-haystack pseudo-Boolean function [39] is highly correlated because it is a constant zero-valued function except for the all-ones binary sequence with co-domain value one. But any EA that does not know the global optimum (i.e. the all-ones sequence) requires exponential expected runtime to find it [3, 39, 70].

Other properties, including deception (where the fitness function misleads EAs away from global optima) [54, 55], modality and isolation (regarding number and distribution of local optima) [147], have been attempted to characterise difficulty of fitness landscapes. However, counterexamples to these properties [92, 114] showed that none of them alone suffices to characterise, in general, the difficulty of fitness landscapes accurately. Or, in other words, each fitness landscape class defined by those properties comprises landscapes of varying, rather than similar, difficulty: a constant landscape and a needle-in-a-haystack landscape are both smooth but the former is trivial and the latter difficult for any EA. Moreover, only a few past studies actually addressed crossover-based fitness landscapes [26, 73, 139], whose search space structure cannot be associated with graphs or neighbourhoods induced by mutation operators [51], which more recent fitness landscape analyses largely overlooked [63, 82, 120] focusing only on mutation-based EAs and local search.

Predictive Measures of Performance

Compared with the previous descriptive approach, characterising fitness landscape classes, a more quantitative approach to analyse fitness landscapes consists in defining statistical measures that estimate how well an EA performs on a given fitness landscape based on specific fitness function values [69, 114].

In the worst case, Jun He and others [62] proved that no general measure, computable in polynomial time, can accurately estimate EA performance for all fitness functions unless P = NP holds [88]. But this does not prevent in theory that useful predictive measures exist for restricted problem classes, and in fact many such measures have been proposed [92]. However, not all of them, including fitness-distance correlation [74] and epistasis variance [29] for GAs, are always reliable measures because they can lead to wrong predictions or have exponential computational cost according to Jansen [69].

Nevertheless, there exist approximate practical measures like accumulated escape probability [120, ch. 5], which does not require prior knowledge of global optima (unlike fitness distance-correlation [74]) and reliably predicted the relative difficulty of several instances of the subset-sum NP-complete problem [76] versus other toy problems (e.g. one-max) against various mutation-only EAs. A more detailed study of problem difficulty of subset-sum, among other binary knapsack NP-hard problems [76], was carried recently by Khulood [82] for local search algorithms.

Furthermore, in contrast to previous approximate and computationally expensive measures [69, 92], Chicano and others [22] showed an exact closed-form expression to compute in polynomial time the auto-correlation (length) [19] for the quadratic assignment problem (including TSP), building upon Weinberger's auto-correlation measure [114, 152] and Stadler's seminal works on elementary landscapes theory [136, 137]. This exact auto-correlation measure, together with local optima network models [120, ch. 9][109], reliably predicted the performance of simulated annealing against several instances of the quadratic assignment problem but less reliably for a steady-state GA using a so-called 'partition' crossover [21].

2.7 Discussion

This chapter broadly reviewed the literature in EC concerning the major families of EAs (Section 2.3) as well as the main theories or frameworks to design EAs (Section 2.4), analyse their search performance (Section 2.5) and fitness landscapes associated with them (Section 2.6). Significant progress has been accomplished in

each of those arenas where frameworks strive for a balance between theoretical rigour and practical utility. But all of them still fall short of one or more of the requirements suggested in Section 2.1 for a general unified theory in EC. That is, a general unified theory should integrate the design and analysis of EAs as well as fitness landscapes within a single cohesive framework. To develop formal taxonomies or hierarchies may be a possible methodology to find a sufficiently general class of EAs with a corresponding sufficiently general class of relevant landscapes while avoiding NFL, provided that many combinatorial optimisation problems actually have sufficient structure that EAs can exploit profitably.

Furthermore, a substantial part of the literature still focuses on a small fraction of all the EA families known (Section 2.3); that is, mutation-only EAs with single-individual populations and GAs for binary string representations. This is mainly due to:

- the behaviour of population-based EAs with crossover is comparably harder to analyse than EAs without populations or without crossover.
- mutation and crossover operators using more complex representations, for instance based on trees as in GP or permutations as in TSP, are harder to analyse and design.
- performance analyses using drift analysis or Markov chain analysis are not easily transferable from one EA-problem pair to another and thus have to be considered on a case-by-case basis.
- the lack of a theoretical basis to analyse how fitness landscapes associated with crossover and other representations than binary strings affect EA performance.

By contrast, a general unified theory of the GF and ELT has the potential to overcome the previous issues, while fulfilling the requirements suggested in Section 2.1, for the reasons which motivated this thesis to begin with (see Section 1.1):

- GF defines a class of crossovers based on a generic notion of metric distance, namely geometric crossovers, independent from specific representations and problems. Geometric crossovers can be designed in a principled manner and cover many existing crossovers for many representations (e.g. sequences, trees, permutations, vectors of reals, etc.) [100].
- GF defines the class of geometric-crossovers EAs as any EA that uses a geometric crossover but no mutation. The abstract population behaviour

of any geometric-crossover EAs is always the same independent of problems and representations: abstract convex evolutionary search [100, 101], which is intuitively described by 'shrinking convex shapes'.

- GF proved a class of geometric-crossover EAs has exponentially better expected runtime than random search on a class of abstract convex landscapes [104], in principle for any problem and representation, without relying on Markov chain analysis nor drift analysis.

- ELT provides a solid theoretical basis to analyse a general class of combinatorial landscapes called elementary landscapes, associated with mutation or a general crossover class known as recombination P-structures, whose fitness function is an eigenvector of a corresponding discrete Laplacian operator [8, 35, 136, 139]. Several real-world problems as well as classic P and NP-complete combinatorial problems have elementary landscapes [56, 85, 91, 121].

Although the unification between GF and ELT is promising, at least the following questions need to be addressed to make their unification viable:

1. What class of crossovers is shared by geometric crossovers and recombination P-structures?

2. What class of EAs doing abstract convex evolutionary search is shared by GF and ELT? Can it use mutation operators?

3. What class of combinatorial landscapes is shared by abstract convex landscapes and elementary landscapes? How does the discrete Laplacian operator corresponding to the shared landscape class relate to the difficulty of problems associated with it?

Part I will address question (1) with a formal classification of crossovers that is missing in the literature [39, 63, 116]. Part II will extend this crossover classification and use it to address question (2), showing how different crossovers in the crossover classification (possibly with mutation) affect the population behaviour described by abstract convex evolutionary search. To do so, Part II will establish links between abstract convexity theory [107, 112, 146] and a generalisation of Holland's schema [65] using Mitavskiy's notion of crossover invariant subsets [98, 99]. Part III will address question (3) by characterising a class of combinatorial landscapes shared by abstract convex landscapes and elementary landscapes, to then show that certain landscapes in such class where certain geometric-crossover EAs expectedly perform well are localised in low-order Laplacian eigenvalues.

Part I

Search Spaces

Abstract

This first part lays the foundation of this thesis based on how the GF and ELT formalise the search space structure induced by mutation or crossover operators, revised next in background Chapters 3 and 4. That is, the set of all possible offspring obtained for given parents after a mutation or crossover operation. By defining search spaces as metric spaces, continuous or finite, the GF generalises traditional mutation and crossovers across representations and problems via metric balls and metric segments (i.e. geodesic intervals). This leads to so-called geometric operators and particularly to geometric crossovers, which cover many crossovers used in practice and can be designed in a principled manner. Geometric mutation and crossovers are seamlessly understood as searching within the same search space. ELT instead adopts a 'one operator one landscape' approach separating mutation search spaces and crossover search spaces, both independent of representations and problems, assuming them finite. The former relies on a simple notion of neighbourhood associated with adjacency matrices of graphs and shortest-path metric distances. The latter are defined in terms of recombination P-structures: a form of finite (non-geodesic) intervals that need not be associated with metrics but can be associated with hypergraphs. In certain cases, a equivalence between mutation and crossover search spaces is possible via a mutation-recombination structure-preserving map.

The contribution of this first part of the thesis is presented in Chapter 5, which focuses on the research question (1) from the literature review in Chapter 2: what class of crossovers is shared by geometric crossovers and recombination P-structures? To address this question, Chapter 5 builds a formal crossover classification upon the class of geometric crossovers and recombination P-structures as well as subclasses thereof defined in Chapters 3 and 4 respectively. Also, a methodology is proposed for how to use or extend the crossover classification. From this crossover classification, geometric recombination P-structures reveal themselves as a general subclass shared by geometric crossovers and recombination P-structures. From Chapter 2, being geometric or highly local are known properties of crossovers favourable to the performance of EAs. However, Chapter 5 proves they do not correspond to identical crossover classes: any geometric crossover is highly local, but not all highly local crossovers are geometric. Finally, Chapter 5 formally explains, in connection with certain inbreeding properties of geometric crossovers, why to classify a crossover as geometric without explicitly knowing the distance could be helpful and the difficulties behind this approach.

Chapter 3

Search Spaces in the Geometric Framework

This chapter revises background material for subsequent chapters. It summarises and clarifies key ideas about the structure of search spaces associated with mutation operators and, especially, crossover operators in the GF. This chapter principally supports the contributions in Chapter 5 on a classification of crossovers that sets the foundation to unify the GF and ELT.

3.1 Search Spaces as Metric Spaces

The GF defines search spaces as *metric spaces*, abstracting any specific problem and algorithmic implementation details, representation of candidate solutions and associated search operators [100]. Candidate solutions become abstract objects in a metric space, and a structure over those objects is established by the distances between them in the metric space. An example is to use a set of Hamming sequences as the solution set and the Hamming distance to impose the structure, resulting in a Hamming graph as the search space. Metric spaces and notions of distance (see Examples 3.1–3.3) are the foundation of the geometric framework (GF).

Definition 3.1 (**Metric space** [94])**.** A *metric space* is a pair (X, d) with an arbitrary set X and a function $d : X \times X \to \mathbb{R}_{\geq 0}$, called *metric* or *distance*, such that $\forall x, y, z \in X$: (I) identity of indiscernibles: $d(x, y) = 0 \iff x = y$; (II) positivity: $d(x, y) > 0$ when $x \neq y$; (III) symmetry: $d(x, y) = d(y, x)$; and (IV) triangle inequality: $d(x, y) \leq d(x, z) + d(z, y)$.

Example 3.1 (**Euclidean metric** [34])**.** The *Euclidean metric* $d_2 : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ outputs the length of the *line segment* between two n-dimensional real vectors: $d_2(x, y) \stackrel{\text{def}}{=} \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$.

Example 3.2 (**Manhattan metric** [34])**.** The *Manhattan* or *rectilinear metric* $d_1 : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ outputs the sum of the lengths obtained by projecting the line segment between two n-dimensional real vectors onto the Cartesian coordinate axes: $d_1(x, y) \stackrel{\text{def}}{=} \sum_{i=1}^{n} |x_i - y_i|$.

Example 3.3 (**Hamming metric** [34])**.** The *Hamming metric* $d_{\text{H}} : \mathcal{H}_q^n \times \mathcal{H}_q^n \to \mathbb{N}_0$ outputs the number of positions in which two sequences $x, y \in \mathcal{H}_q^n = \{0, 1, \ldots, q-1\}^n$ differ: $d_{\text{H}}(x, y) \stackrel{\text{def}}{=} |\{i : 1 \leq i \leq n, x_i \neq y_i\}|$.

Definition 3.1 makes clear that candidate solutions are points in a metric space and its metric defines the structure over them, but what about the search operators? The two prominent kinds of search operators in EAs that generate new candidate solutions are mutation and crossover (or recombination), and the GF models them across representations and problems using two abstractions in metric spaces, namely *metric balls* and *segments* (Section 3.2). Briefly, the GF's approach is to choose first a metric space for the problem at hand, which comes with such abstract mutation and crossover, and then derive from these abstractions the corresponding specific operators that would be actually implemented for that metric space. A major feature of this approach is that metric spaces (i.e. search spaces) are decoupled to an extent from specific search operators, thus enabling distinct search operators to search within the same metric space, hence within the same search space. Section 3.2 and Section 3.3 explain what is the relationship between metric spaces, the abstractions over mutation and recombination, and the actual mutation and recombination operators used in EAs.

3.2 Geometric Mutation and Crossover

To abstractly define mutation and crossover in metric spaces, the GF uses metric balls and metric segments.

Definition 3.2 (**Metric ball and segment** [146])**.** Let (X, d) be any metric space. A *closed ball* centred at point $x \in X$ with radius $r \in \mathbb{R}_{\geq 0}$ is defined as $\bar{B}_d(x, r) \stackrel{\text{def}}{=} \{y \in X \mid d(x, y) \leq r\}$. A *geodesic interval* or *metric segment* is defined as $[x, y]_d \stackrel{\text{def}}{=} \{z \in X \mid d(x, z) + d(z, y) = d(x, y)\}$, where $x, y \in X$ are called the *extremes* of the segment and $d(x, y)$ its *length*.

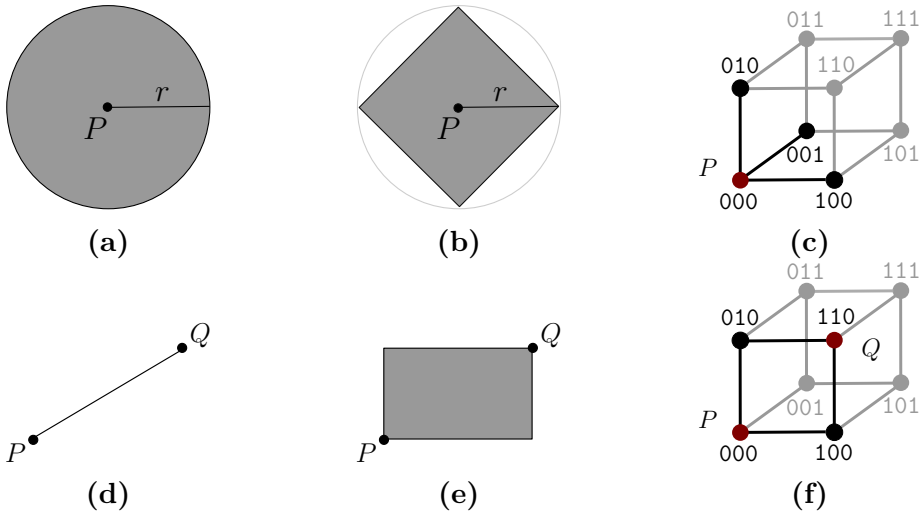


Figure 3.1. Example of a metric ball and a metric segment for the Euclidean metric space $(\mathbb{R}^2, d_2)$ in Figures 3.1a and 3.1d, Manhattan metric space $(\mathbb{R}^2, d_1)$ in Figures 3.1b and 3.1e, and Hamming metric space $(\mathcal{H}_2^3, d_{\mathrm{H}})$ in Figures 3.1c and 3.1f.

Observe in Figures 3.1a–3.1c that a metric ball captures the notion of neighbourhood around a point, characteristic of mutation (e.g. single-bit flip), and also that a metric segment captures the notion of picking two parents and returning a set of offspring (Figures 3.1d–3.1f), characteristic of crossover (e.g. one-point). Based on them, abstract mutation and crossover are defined geometrically via metrics, hence the name *geometric operators* (Definitions 3.3–3.4).

GF emphasises the structure of geometric operators over other aspects such as probability distributions imposed on offspring sets. The reason behind is the GF prioritises the underlying search space structure associated with search operators over their probabilistic nature [100]. This is possible if one considers the *support sets* of search operators: the sets of all immediate offspring in the search space reachable from given parents, namely offspring generated with non-zero probability, which are fixed once a search operator is chosen. For simplicity, this thesis does not make use of probability distributions over metric spaces, for this entails extending the GF to metric measure spaces [101, 146], and that is out of scope. Also, this thesis follows the same representation-independent setting adopted in GF, which is transparent to whether genotype-phenotype maps [123] are explicitly defined or not.

Definition 3.3 (**[Complete/Incomplete] geometric crossover** [100])**.** Let (X, d) be any metric space. An operator $\xi : X \times X \to \mathcal{P}(X)$ is a *geometric crossover* for (X, d), if all offspring support sets belong to the metric segment between parents. That is, if $\xi(x, y) \subseteq [x, y]_d$ for all parents $x, y \in X$. If $\forall x, y \in X : \xi(x, y) = [x, y]_d$, then ξ is a *complete geometric crossover*. If ξ is not complete, it is called *incomplete*

geometric crossover. The *class* of geometric crossovers is denoted $\mathcal{GX}$, and $\mathcal{GX}_{fin}$ for finite geometric crossovers (i.e. with X finite). The *subclasses* of complete, incomplete, finite-complete, and finite-incomplete geometric crossovers are denoted respectively: $\mathcal{GX}$-complete, co-$\mathcal{GX}$-complete $\stackrel{\text{def}}{=} \mathcal{GX} \setminus \mathcal{GX}$-complete, $\mathcal{GX}$-complete$_{fin}$ and co-$\mathcal{GX}$-complete$_{fin}$ $\stackrel{\text{def}}{=} \mathcal{GX}_{fin} \setminus \mathcal{GX}$-complete$_{fin}$.

Definition 3.4 (**Geometric mutation** [100])**.** Let (X, d) be any metric space. An operator $\mu_\varepsilon : X \to \mathcal{P}(X)$ is a *geometric mutation* for (X, d), if all offspring support sets of μ_ε for an individual belong to a metric closed ball around such individual. That is, if $\forall x \in X : \mu_\varepsilon(x) \subseteq \bar{B}_d(x, \varepsilon)$ where $\varepsilon \in \mathbb{R}_{\geq 0}$ is the smallest non-negative real number for which that condition holds and $\varepsilon = 1$ is assumed if unspecified.

From Definitions 3.3–3.4 it is clear that geometric crossover and mutation search in the same metric space (hence same search space) after one is fixed, even if the specific search operators they represent are different. For instance, even though Hamming balls and segments (Figures 3.1c and 3.1f respectively) represent single-bit flip mutation and unbiased uniform crossover, Hamming balls and segments are well defined for the same Hamming graph.

Definition 3.5 (**Uniform crossover**)**.** Let $(\mathcal{H}_q^n, d_{\text{H}})$ be any n-dimensional q-ary Hamming metric space with $q \geq 2$. The *support function* of the traditional (unbiased) uniform crossover is the function $\textsc{Uniform} : \mathcal{H}_q^n \times \mathcal{H}_q^n \to \mathcal{P}(\mathcal{H}_q^n), (x, y) \mapsto \{(z_1, \ldots, z_n) \mid z_i \in \{x_i, y_i\},\ 1 \leq i \leq n\}$ for any parents $x, y \in \mathcal{H}_q^n$.

Example 3.4 (**Finite-complete geometric: uniform crossover**)**.** The traditional (unbiased) uniform crossover on a finite set of Hamming sequences is a finite-complete geometric crossover because its support set coincides with the Hamming metric segment between parents. For instance, consider parents $\texttt{000}, \texttt{110} \in \mathcal{H}_2^3$. Then, $\textsc{Uniform}(\texttt{000}, \texttt{110}) = [\texttt{000}; \texttt{110}]_{d_{\text{H}}} = \{\texttt{000}, \texttt{010}, \texttt{100}, \texttt{110}\}$. Analogously for any other pair of parents [100].

Definition 3.6 (**Asymmetric one-point crossover**)**.** Let $(\mathcal{H}_q^n, d_{\text{H}})$ be any n-dimensional q-ary Hamming metric space with $q \geq 2$. Let any two parents $x, y \in \mathcal{H}_q^n$ and a crossover point $i \in \{1, \ldots, n\}$. The *asymmetric one-point crossover* of x and y at i returns a single offspring z formed by concatenating all symbols of x up to position i with all symbols (if any) of y in remaining positions. Formally: $\text{asym-}\textsc{OnePoint}(i) : \mathcal{H}_q^n \times \mathcal{H}_q^n \to \mathcal{H}_q^n,\ (x, y) \mapsto z = (z_j)_{j \in \{1,\ldots,n\}}$, where

$$z_j = \begin{cases} x_j, & \text{if } j \leq i, \\ y_j, & \text{if } (i < j) \text{ and } (i \neq n). \end{cases}$$

The *support function* of asymmetric one-point crossover is: asym-OnePoint$(x, y) \stackrel{\text{def}}{=} \bigcup_{1 \leq i \leq n}$ asym-OnePoint$(i)(x, y)$.

Example 3.5 (**Finite-incomp. geometric: asymmetric one-point crossover**)**.** The asymmetric one-point crossover on a finite set of Hamming sequences is a finite-incomplete geometric crossover because its support set may not coincide with the Hamming metric segment between parents. For instance, asym-OnePoint$(\mathtt{111}, \mathtt{000})$ $= \{\mathtt{100}, \mathtt{110}, \mathtt{111}\} \not\ni \mathtt{011} \in [\mathtt{111}, \mathtt{000}]_{d_{\mathrm{H}}} = \{\mathtt{0}, \mathtt{1}\}^3$.

Remark 3.1 (Definition 3.3)**.** To be geometric, a crossover only requires a single instance of a metric space fulfilling Definition 3.3. But there is a caveat for generic metric spaces (e.g. n-dimensional q-ary Hamming metric spaces): if a crossover is geometric, it must be so for every instance (i.e. every dimension n and alphabet size q).

Abstracting mutation and recombination in this geometric fashion has the following three main advantages [100].

First, it enables mutation and recombination to be described within the same search space, which historically has not always been the case. For instance, under the 'one operator one landscape' view of Jones [73], a GA searches not in one but several search spaces simultaneously (one when doing mutation, other for recombination, other for selection); and, Gitchoff, Stadler and Wagner [51, 139] use graphs to formalise search spaces associated with mutation but recombination involves distinct structures based on hypergraphs [6].

Secondly, geometric operators can help us formalise what is a mutation and recombination operator in problems for new or unfamiliar search spaces (e.g. trees, variable-size sets). This is possible because: metric balls and segments are not tied to a specific problem or solution representation, and there exist a natural duality between edit distances [34] and syntactic operations on sequences, trees, permutations, etc. [100].

Thirdly, the definition of geometric crossover (and mutation) may be interpreted as a 'template' that can be instantiated by choosing a specific distance to obtain a notion of geometric crossover tailored to such metric space. Effectively, the definition of geometric crossover offers a strategy to design crossovers (see next Section 3.3).

Geometric operators attain as much generality as metric spaces allow. Nevertheless, not all crossovers are compatible with metric spaces (i.e. their structure cannot be expressed via metrics) such as the unequal crossover on strings [131, 135] and Koza's subtree crossover in GP [86, 103]. Sections 3.4–3.5 provide more detail on non-geometric crossovers.

3.3 Principled Design of Geometric Crossovers

Crossovers operators are abstracted from their specific definitions when their structure is formalised via metric segments (i.e. geodesic intervals), obtaining thus the abstract notion of geometric crossover (Section 3.2). Loosely speaking, *principled design of geometric crossovers*[1] [100] consists in 'reversing' such process of abstraction, summarised as follows.

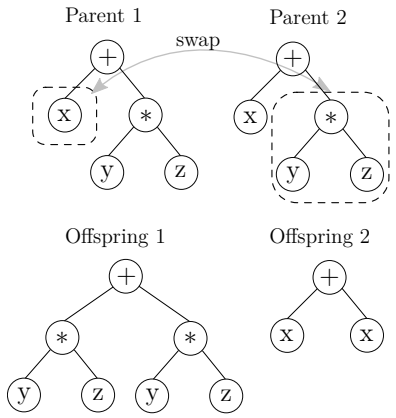
Geometric crossovers are defined by metric segments (Definition 3.3), which is a function parametrised by a generic metric distance. Substituting a specific distance results in distinct and specific kinds of metric segments (Figures 3.1d–3.1f), and each delimits a corresponding notion of geometric crossover. In theory, there are as many valid notions of geometric crossover as distances. In practice, one needs a specific crossover operator to implement it on a digital computer, and not all distances may be adequate to obtain a geometric crossover. For example, geometric crossovers under the reversal distance, based on the inversion or 2-opt operator [4], cannot be efficiently implemented in an exact manner, only approximately [100]. The GF uses the following criteria for what is a 'good' distance: (a) a distance d for which there exists an efficient implementation of a crossover that is geometric under d; and, (b) a distance d for which a given fitness function defined on a metric space (X, d) induces a fitness landscape 'easy' to search using the crossover operator in (a).

Depending on what is known about a particular given problem, finding such distance may be more or less difficult for the designer. Here the GF follows the premise 'good mutation, good crossover' [100]: if one knows a particular mutation operator performs well, and it is associated with a distance, then the same distance can be used to define a specific notion of geometric crossover that expectedly performs well. This rule-of-thumb assumes that a 'good' neighbourhood structure given by mutation is likely to perform well for different meta-heuristics [52].

3.4 Non-geometric Crossovers

Many crossovers are geometric for a wide variety of search spaces (e.g. trees, permutations, sequences, etc.), but not all of them are geometric. Two examples are Koza's subtree swap [86] and Davis's order [31] crossovers (Figure 3.2).

[1] Principled design also applies to geometric mutation (Definition 3.4). In fact, the design of operators based on distances is not necessarily limited to geometric operators: geometric shapes other than metric balls and segments may be considered to define other forms of principled design. This thesis does not consider such other forms of principled design.



Parent 1 : 1 2 3 (Crossover section: 2)
Parent 2 : 2 3 1 (Available fillers: 3, 1)
Offspring 1 : 3 2 1

Offspring 1 : 3 2 1 (Crossover section: 1)
Parent 1 : 1 2 3 (Available fillers: 2, 3)
Parent 2 : 2 3 1

Figure 3.2. Examples of non-geometric crossovers: Koza's subtree swap ('left') and Davis's order ('right'). They are not geometric since they fail, respectively, the purity and convergence inbreeding properties of geometric crossovers (Section 3.5).

Definition 3.7 (**Non-geometric crossover** [103])**.** Let X be an arbitrary set and $D \stackrel{\text{def}}{=} \{d \mid d \in \mathbb{R}^{(X\times X)}, d \text{ is a metric}\}$ the set of all metrics on X. A crossover ξ is a *non-geometric crossover* if, for every metric, there always exists at least one offspring that does not belong to the metric segment between parents. That is, if $\forall d \in D\ \exists x, y, z \in X : z \in \xi(x, y)$ and $z \notin [x, y]_d$.

Proposition 3.1 (**Existence of non-geometric crossovers** [100, 103])**.** The class of non-geometric crossovers $\overline{\mathcal{GX}}$ is not empty. Therefore, recombination operators split into two classes: geometric $\mathcal{GX}$ and non-geometric $\overline{\mathcal{GX}}$.

Remark 3.2 (Proposition 3.1)**.** The class of non-geometric crossovers $\overline{\mathcal{GX}} \stackrel{\text{def}}{=} \mathcal{U}\backslash\mathcal{GX}$ is the complement of the geometric class $\mathcal{GX}$ with respect to the universal class of all crossovers $\mathcal{U}$.

A lesson from the no-free-lunch (NFL) theorems [68, 155] is that to provide reliable performance assurances for combinatorial optimisation algorithms, deterministic or stochastic (like most EAs), it is necessary to embed knowledge about the structure of the problem's objective function into the algorithm. So that the algorithm can exploit it during the search (i.e. the algorithm must 'match' the class of problems it wants to solve). Otherwise it is not statistically guaranteed to perform better than pure random search. In short, restricted problem-algorithm class pairs must be considered.

In the light of NFL, Proposition 3.1 (see Theorem 14.4.4 in [100]) is a fundamental result of the GF because it states that the GF is not a theory of all crossovers (hence the class of EAs based on geometric crossovers is effectively restricted), and because the practitioner can embed problem knowledge through a suitably chosen

metric when designing an EA using geometric crossovers [100]. By problem knowledge, the GF means that the fitness landscape fulfils certain conditions[2], namely certain kinds of smoothness and abstract convexity, defining a class of landscapes where geometric-crossover EAs (without mutation) can outperform pure random search [101, 104]. By suitable metric, the GF means that the metric should be chosen carefully because such classes of fitness landscape and geometric-crossover EA may or may not match depending on which metric is chosen. In short, Proposition 3.1 itself does not say that the GF is a 'useful' theory, but it does say that the GF is not a futile theory inasmuch as it escapes the NFL negative consequences.

Chapter 9 discusses abstract convexity conditions of fitness landscapes where geometric-crossover EAs with geometric crossover and no mutation expectedly outperform pure random search.

3.5 Inbreeding Properties of Geometric Crossovers

Geometric crossovers are a relevant class of crossover operators: many crossovers used in practice are geometric and can be designed systematically across representations (Sections 3.2–3.4). To distinguish which crossovers are non-geometric is equally important, and this is the purpose of the inbreeding properties described next.

To prove that a crossover is geometric, it suffices to find at least one metric where Definition 3.3 holds, which can be challenging. Proving that a crossover is non-geometric is significantly harder, if not nearly impossible, for one may have to test an arbitrarily large (possibly infinite) number of metrics before reaching a conclusion according to Definition 3.7. That is not feasible. The *inbreeding properties* of geometric crossovers stated in Propositions 3.2–3.4 (see Theorems 1–3 in [103]) circumvent such complication. The inbreeding properties (Figure 3.3) are independent of any underlying metric space because they do not involve distances, and they provide sufficient conditions to determine if a crossover is non-geometric. It suffices to find a single counterexample where any of them fails to conclude automatically that such crossover is not geometric for every metric space, so that one need not test arbitrarily many metrics. An open problem is whether the inbreeding properties [103] are also sufficient conditions to guarantee geometricity of crossovers: if a crossover fulfils all three, is it geometric for some metric?

[2]The GF assumes that the designer can derive or has access to the closed mathematical expression of the fitness function and metric.

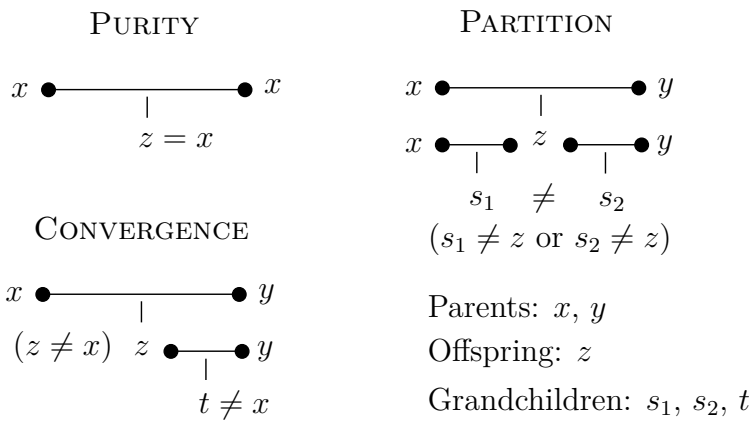


Figure 3.3. Diagrams of the inbreeding properties of geometric crossovers. *Purity*: mating two identical parents x produces an identical offspring $z = x$. *Convergence*: if parents x and y produce an offspring z such that $z \neq x$, and z mated with y produces a grandchild t, then $t \neq x$. *Partition*: if parents x and y produce an offspring z which in turn is mated separately with both parents leading to grandchildren s_1 and s_2, then $s_1 \neq s_2$ when at least one of the grandchildren is different from z.

Proposition 3.2 (**Property of purity** [103])**.** Let ξ be a binary crossover operator defined on a set X. If ξ is a geometric crossover for some unspecified metric on X, then recombining an individual with itself can only produce itself as offspring. That is, $\forall x \in X : \xi(x, x) = \{x\}$.

Proposition 3.3 (**Property of convergence** [103])**.** Let ξ be a binary crossover operator defined on a set X, and offspring $z \in \xi(x, y)$ for any parents x and y. If ξ is a geometric crossover for some unspecified metric on X, then $\xi(x, z)$ cannot produce y unless $z = y$. That is, $\forall x, y \in X$ and $\forall z \in \xi(x, y) : y \in \xi(x, z) \implies z = y$. Analogously, $\xi(z, y)$ cannot produce x unless $z = x$.

Proposition 3.4 (**Property of partition** [103])**.** Let ξ be a binary crossover operator defined on a set X, and any parents $x, y \in X$. Let also a child $z \in \xi(x, y)$, and grandchildren $s_1 \in \xi(x, z)$ and $s_2 \in \xi(z, y)$ such that $s_1 \neq z$ or $s_2 \neq z$ (or both). If ξ is a geometric crossover for some unspecified metric on X, then $s_1 \neq s_2$. In other words, z is the only offspring (if any) common to $\xi(x, z)$ and $\xi(z, y)$.

In summary, the main relationships between geometric crossovers and the inbreeding properties are: (a) if ξ is a geometric crossover for some metric space, then ξ fulfils the inbreeding properties; and, (b) if a crossover ξ fails an inbreeding property, then ξ is non-geometric for every metric space.

Chapter 4

Search Spaces in Elementary Landscapes Theory

This chapter revises background material for subsequent chapters. It summarises and clarifies key ideas about the structure of search spaces associated with mutation and recombination operators in ELT. This chapter principally supports the contributions in Chapter 5 on a crossover classification that sets the foundation to unify GF and ELT.

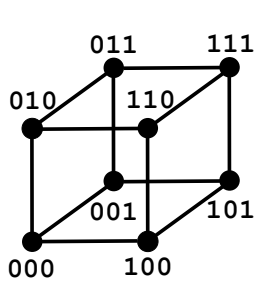
4.1 Mutation Search Spaces

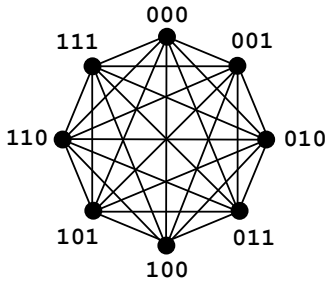
Mutation operators are normally implemented as probabilistic algorithms, but it is possible to define a corresponding search space structure that is fixed by means of their *support functions* and *support sets*. A support set of a mutation operator is a set of all possible candidate solutions in the search space that result from applying the mutation operator to a given candidate solution. Support sets are determined by a support function called *neighbourhood function* $N : X \to \mathcal{P}(X)$ mapping a candidate solution $x \in X$ to a subset $N(x) \in \mathcal{P}(X)$ of neighbouring solutions in the power set of all candidate solutions X. Since neighbourhoods and their supports sets are fixed, this approach allows ELT [136] to formalise a *mutation search space* as a fixed pair (X, N) consisting of a domain X (i.e. the set of all candidate solutions) and a neighbourhood function N imposing a structure on X. Hence the mutation search space is determined as soon as the mutation operator and domain X are chosen. ELT does not make particular assumptions about genotype-phenotype maps or solution representations.

Elementary landscapes theory (ELT) focuses on combinatorial optimisation prob-

lems whose search spaces are finite as opposed to continuous spaces like Euclidean spaces $\mathbb{R}^n$. Accordingly, ELT uses *graphs* to describe finite mutation search spaces (X, N), where each vertex describes a candidate solution and each edge a neighbour relationship between the two end vertices of the edge. This is possible due to a natural equivalence between adjacency matrices of graphs and neighbourhoods: a vertex y is adjacent to x if and only if y neighbours x. In other words, graphs formalise search spaces associated with mutation operators because adjacency matrices, like neighbourhoods, define supports sets of mutation operators. To illustrate it, consider the (generalised) multiple-bit flip neighbourhood for general Hamming sequences.

Definition 4.1 (**Multiple-bit flip neighbourhood**)**.** Let $\mathcal{H}_q^n$ be n-dimensional Hamming sequences with alphabet size $q \geq 2$. The (generalised) *multiple-bit flip neighbourhood*, parametrised by k such that $1 \leq k \leq n$, is defined by $\textsc{Bitflip}(k) : \mathcal{H}_q^n \to \mathcal{P}(\mathcal{H}_q^n)$, $x \mapsto \{y \in \mathcal{H}_q^n \mid d_{\mathrm{H}}(x, y) \leq k,\ x \neq y\}$. So that $y \in \textsc{Bitflip}(k)(x)$ (i.e. y neighbours x) if their Hamming distance $d_{\mathrm{H}}(x, y)$, namely the number of positions in which both sequences differ, never exceeds the maximum k.





$$\mathbf{A}(H(3,2)) = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

(a)

$$\mathbf{A}(K_8) = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

(b)

Figure 4.1. The hypercube graph $H(3, 2)$ in Figure 4.1a and the complete graph K_8 in Figure 4.1b, labelled using binary sequences, with their respective adjacency matrices.

For n-dimensional binary Hamming sequences $\mathcal{H}_2^n$, $\textsc{Bitflip}(k)$ is the neighbourhood associated with the supports sets of traditional (multiple) bit flip mutation

operators.

For example, consider three-dimensional binary Hamming sequences $\mathcal{H}_2^3$ and single-bit flip mutation, which selects one bit at random from a given binary sequence and applies bitwise negation to the selected bit. Then, its neighbourhood is Bitflip(1). So $\texttt{001} \in \textsc{Bitflip}(1)(\texttt{000})$ because 001 is reachable from 000 in a single bit flip. Equivalently, $\{\texttt{000}, \texttt{001}\}$ is an edge of the three-dimensional binary Hamming graph $H(3, 2)$ or hypercube. Therefore, the search space associated with Bitflip(1) is a hypercube (Figure 4.1a).

Now, consider three-bit flip mutation[1], which selects three bits at random from a given binary sequence and applies bitwise negation to the three selected bits. Its neighbourhood is Bitflip(3). So $\texttt{111} \in \textsc{Bitflip}(3)(\texttt{000})$ because 111 is reachable from 000 in three bit flips. Equivalently, $\{\texttt{000}, \texttt{111}\}$ is an edge of the complete graph K_8 with vertex set all three-dimensional binary Hamming sequences $\mathcal{H}_2^3$. In other words, the search space associated with Bitflip(3) is the complete graph K_8 in Figure 4.1b because every sequence is reachable from any given sequence under a single application of three-bit flip mutation.

Graphs may be of various kinds depending on their adjacency matrices (or neighbourhoods): regular or irregular, undirected or directed, unweighted or weighted. This thesis, unless stated otherwise, assumes the 'simplest' scenario addressed by ELT: a graph[2] without loops, regular, undirected and unweighted [136]. Extensions to ELT for search spaces with irregular or directed graphs are possible [5, 35], but this thesis does not focus on them because ELT is less developed in such cases.

4.2 Recombination Search Spaces

Research in ELT on recombination operators [139, 140, 150] develops as an extension to ELT's original work on mutation [136] rather than as part of its foundation. This led to formalise recombination search spaces differently from mutation search spaces and in a way that complicates their comparison (Sections 4.2.1–4.2.2).

[1] A three-bit flip mutation is also achieved with three consecutive single-bit flips on the same individual. However, that does not mean their neighbourhoods are identical. For three-bit flip mutation 111 neighbours 000, whereas for single-bit flip mutation 111 does not neighbour 000.

[2] The conditions of regularity and undirectedness of a graph are expressed via neighbourhoods by requiring respectively that: all individuals have the same number of neighbours, namely $|N(x)|$ must be constant, and the neighbourhood is symmetric, that is $y \in N(x) \iff x \in N(y)$.

4.2.1 Recombination P-structures

Similar to mutation operators (Section 4.1), a fixed search space structure can be defined for recombination operators via their *support functions* and *support sets*. A support set of a recombination operator is a set of all possible candidate solutions in the search space that result from applying the recombination operator to a given pair of candidate solutions (i.e. parents).

Unlike mutation operators, recombination operators act on two individuals, which complicates formalising their support functions and corresponding search space. Here, neighbourhood functions or adjacency matrices (Section 4.1) are inappropriate because neighbourhoods are not binary functions defined on pairs of individuals, and adjacency relations are vertex-to-vertex where each vertex represents at most one individual not two.

Instead, ELT relies on the notion of interval in *interval spaces* (Definition 4.2), where endpoints of the interval would represent parents and points 'in between' would represent offspring. Formally, that is an *interval operator* $I : X \times X \to \mathcal{P}(X)$ mapping a pair of (parent) candidate solutions to a set of (offspring) candidate solutions in the power set $\mathcal{P}(X)$ of the search space domain X.

Definition 4.2 (**Interval space** [146])**.** Let $I : X \times X \to \mathcal{P}(X)$ be any function defined on any set X such that $\forall x, y \in X$: (I) extensitivity: $x, y \in I(x, y)$; and, (II) symmetry: $I(x, y) = I(y, x)$. Then, I is an *interval operator* on X, $I(x, y)$ is the *interval between* x and y, and (X, I) is an *interval space*.

A *recombination search space* is formalised in ELT as a particular case of an interval space known as *recombination P-structure* (Definition 4.3). So all examples of recombination P-structures seen in this and forthcoming chapters are interval spaces. Besides intervals, recombination P-structures have been similarly described as *transit functions* [18], introduced in [107] to unify intervals, paths, convex structures and betweenness relations in graphs and partially ordered sets [17, 112, 146].

Definition 4.3 (**Recombination P-structure** [51, 139])**.** Let X be any non-empty finite set and any interval operator $\mathcal{R} : X \times X \to \mathcal{P}(X)$. Then, $(X, \mathcal{R})$ is a *recombination P-structure*, if $\forall x, y, z \in X$: (I) fix-point: $\mathcal{R}(x, x) = \{x\}$; (II) symmetry: $\mathcal{R}(x, y) = \mathcal{R}(y, x)$; (III) null-recombination: $\{x, y\} \subseteq \mathcal{R}(x, y)$; and (IV) size-monotonicity: if $z \in \mathcal{R}(x, y)$, then $|\mathcal{R}(x, z)| \leq |\mathcal{R}(x, y)|$. The *class*[3] of all interval operators $\mathcal{R}$ of recombination P-structures is denoted $\mathcal{RP}$.

[3] Abusing the language, if no confusion arises, the 'class of recombination P-structures' refers hereinafter indifferently to the class of interval operators $\mathcal{R}$ or recombination P-structures $(X, \mathcal{R})$.

Remark 4.1 (Definition 4.3)**.** Recombination P-structures $(X, \mathcal{R})$ may be considered independent of a specific optimisation problem and representation of solutions as far as the finite domain X where they are defined is independent. Recombination P-structures cannot be used for problems defined on continuous domains, like Euclidean spaces, unless the finiteness of X is relaxed.

One of the simplest examples of a recombination P-structure, defined for any finite set of candidate solutions X, is $(X, \textsc{Id})$ given by the support function of the *identity crossover* $\textsc{Id}$ (Definition 4.4). More interesting recombination P-structures are those associated with uniform and one-point crossovers in Examples 4.1–4.2.

Definition 4.4 (**Identity crossover** [139])**.** Let X be a non-empty finite set. The support function of *identity crossover* is $\textsc{Id}(x, y) \stackrel{\text{def}}{=} \{x, y\}$ for any parents $x, y \in X$.

Example 4.1 (**Uniform recomb. P-structure** [51, 139])**.** Recall Definition 3.5 of the support function $\textsc{Uniform}$ for the (unbiased) uniform crossover on n-dimensional q-ary ($q \geq 2$) Hamming sequences $\mathcal{H}_q^n$. Then, $\mathscr{R}_\Omega \stackrel{\text{def}}{=} (\mathcal{H}_q^n, \textsc{Uniform})$ is the uniform recombination P-structure. For instance, verifying the recombination P-structure axioms for parents `100` and `001` yields:

(I) $\textsc{Uniform}(\texttt{100}, \texttt{100}) = \{\texttt{100}\}$, and $\textsc{Uniform}(\texttt{001}, \texttt{001}) = \{\texttt{001}\}$;

(II) $\textsc{Uniform}(\texttt{100}, \texttt{001}) = \textsc{Uniform}(\texttt{001}, \texttt{100}) = \{\texttt{100}, \texttt{001}, \texttt{000}, \texttt{101}\}$;

(III) $\{\texttt{100}, \texttt{001}\} \subseteq \textsc{Uniform}(\texttt{100}, \texttt{001})$;

(IV) $|\textsc{Uniform}(\texttt{100}, \texttt{000})| = 2 \leq 4 = |\textsc{Uniform}(\texttt{100}, \texttt{001})|$.

Definition 4.5 (**One-point crossover**)**.** Recall from Definition 3.6 the asymmetric one-point function asym-$\textsc{OnePoint}(i)$ at crossover point i on n-dimensional q-ary ($q \geq 2$) Hamming sequences $\mathcal{H}_q^n$. The *one-point function* parametrised with crossover point $1 \leq i \leq n$ is the 'symmetrised' version of asym-$\textsc{OnePoint}(i)$:

$$\begin{aligned} \textsc{OnePoint}(i) : \mathcal{H}_q^n \times \mathcal{H}_q^n &\to \mathcal{H}_q^n \times \mathcal{H}_q^n \\ (x, y) &\mapsto (s, t) = \{\text{asym-}\textsc{OnePoint}(i)(x, y), \\ &\qquad \text{asym-}\textsc{OnePoint}(i)(y, x)\} \end{aligned}$$

for any parents (x, y) and offspring (s, t). The *support function* of the traditional one-point crossover is $\textsc{OnePoint}(x, y) \stackrel{\text{def}}{=} \bigcup_{1 \leq i \leq n} \textsc{OnePoint}(i)(x, y)$.

Example 4.2 (**One-point recomb. P-structure** [51, 139])**.** Let n-dimensional q-ary ($q \geq 2$) Hamming sequences $\mathcal{H}_q^n$. Then, $\mathscr{R}_1 \stackrel{\text{def}}{=} (\mathcal{H}_q^n, \textsc{OnePoint})$ is the one-point

recombination P-structure. For instance, verifying the recombination P-structure axioms for parents 000 and 111 yields:

(I) $\textsc{OnePoint}(000, 000) = \{000\}$, and $\textsc{OnePoint}(111, 111) = \{111\}$;

(II) $\textsc{OnePoint}(000, 111) = \textsc{OnePoint}(111, 000) = \{0, 1\}^3 \setminus \{010, 101\}$;

(III) $\{000, 111\} \subseteq \textsc{OnePoint}(000, 111)$;

(IV) $|\textsc{OnePoint}(000, 011)| = 4 \leq 6 = 2^3 - 2 = |\textsc{OnePoint}(000, 111)|$.

Remark 4.2 (Example 4.2)**.** Notice that $(\mathcal{H}_q^n, \text{asym-}\textsc{OnePoint})$ is not a recombination P-structure since asym-OnePoint (Definition 3.6) fails the symmetry axiom. For instance, $\text{asym-}\textsc{OnePoint}(111, 000) \not\ni 011 \in \text{asym-}\textsc{OnePoint}(000, 111)$.

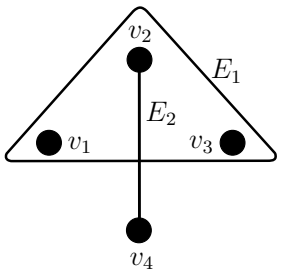
Remark 4.3 (Definition 4.3)**.** If $(X, \mathcal{R})$ is a recombination P-structure and the set X is parametrised, say n-dimensional Hamming sequences $\mathcal{H}_q^n$ for alphabet size q, then $\mathcal{R}$ must fulfil the recombination P-structure axioms for all values of n and q.

Thus far the situation is mutation and recombination search spaces are formally distinct in ELT, graphs represent the former and recombination P-structures the latter. This disagreement may be interpreted from the 'one operator one landscape' view of Jones [73] by focusing on the support structure of mutation and recombination operators and voluntarily ignoring their transition probabilities from parents to offspring. That is, because neighbourhoods (i.e. support functions of mutation operators) and abstract intervals (i.e. support functions of recombination operators) are formally distinct, so are their associated search spaces: each search operator induces its own search space [73]. Nevertheless, such disagreement is true only to a certain point [26, 51, 139]. Next, Section 4.2.2 explains in what sense mutation and recombination search spaces are equivalent in ELT.

4.2.2 Hypergraphs, Backbone Graphs and the Mutation-recombination Homomorphism

Mutation and recombination are the main operators driving evolutionary change in EAs. Consequently, knowing if and how mutation and recombination structures are different is key to understand the effect of these operators on EAs. Mutation affords a simple and natural definition via graphs (Section 4.1) but recombination, namely recombination P-structures, does not (Section 4.2.1). This motivates Gitchoff and Wagner [51] to suggest hypergraphs.

A *hypergraph* G is a generalisation of a graph whose edges (alias *hyperedges*) are formed by a set of vertices rather than just a pair of vertices (Figure 4.2). Formally, it is a pair $(V(G), E(G))$, or simply (V, E) if G is clear from the context, that consist of a finite vertex set $V = \{v_1, v_2, \ldots, v_n\}$ and a family $E = \{E_1, E_2, \ldots E_m\}$ of subsets of V, its hyperedges, such that: (a) $E_j \neq \emptyset$, for all $j \in \{1, \ldots, m\}$; and, (b) $\cup_{j=1}^{m} E_j = V$. A hypergraph G is also defined by its binary *incidence matrix* $\mathbf{H} = (h_{i,j}) \in \{0, 1\}^{n \times m}$ whose rows $h_{i,\cdot}$ are indexed with vertices v_i and columns $h_{\cdot,j}$ with hyperedges E_j, where $h_{i,j} = 1$ if $v_i \in E_j$ ($h_{i,j} = 0$ otherwise) [6].



$$\mathbf{H} = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cc} E_1 & E_2 \end{array} \\ \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \end{array}$$

Figure 4.2. A hypergraph ('left') with vertex set $V = \{v_1, v_2, v_3, v_4\}$, hyperedge set $E = \{E_1, E_2\}$ where $E_1 = \{v_1, v_2, v_3\}$ and $E_2 = \{v_2, v_4\}$, and incidence matrix $\mathbf{H}$ ('right').

The hypergraph of a recombination P-structure $(X, \mathcal{R})$ has vertex set X (i.e. the set of candidate solutions) and its hyperedge set is the family of all possible offspring sets $\mathcal{R}(x, y)$. So hyperedges are the support sets $\mathcal{R}(x, y)$ of the recombination operator with support function $\mathcal{R}$.

Definition 4.6 (**Recomb. P-structure hypergraph** [139])**.** Denote $\mathscr{R}$ a recombination P-structure $(X, \mathcal{R})$. Its *hypergraph* is $\text{hyp}_{\mathscr{R}} \stackrel{\text{def}}{=} \big(V(\text{hyp}_{\mathscr{R}}), E(\text{hyp}_{\mathscr{R}})\big)$ where $V(\text{hyp}_{\mathscr{R}}) = X$ is the vertex set, and the hyperedge set is $E(\text{hyp}_{\mathscr{R}}) = \{\mathcal{R}(x, y) \mid x, y \in V(\text{hyp}_{\mathscr{R}}), \mathcal{R}(x, y) \neq \emptyset\}$. Each offspring set $\mathcal{R}(x, y)$ is a hyperedge.

Besides Definition 4.6, hypergraphs of recombination P-structures are defined through their binary incidence matrices $\mathbf{H}$ where each entry $h_{x,(y,z)}$ indicates whether or not x is offspring of parents y and z under $\mathcal{R}$

$$h_{x,(y,z)} \stackrel{\text{def}}{=} \begin{cases} 1, & \text{if } x \in \mathcal{R}(y,\, z), \\ 0, & \text{otherwise.} \end{cases} \tag{4.1}$$

Rows $h_{x,\cdot}$ are indexed by all offspring $x \in X$ and columns $h_{\cdot,(y,z)}$ by all pairs of parents $(y, z) \in X \times X$ [139].

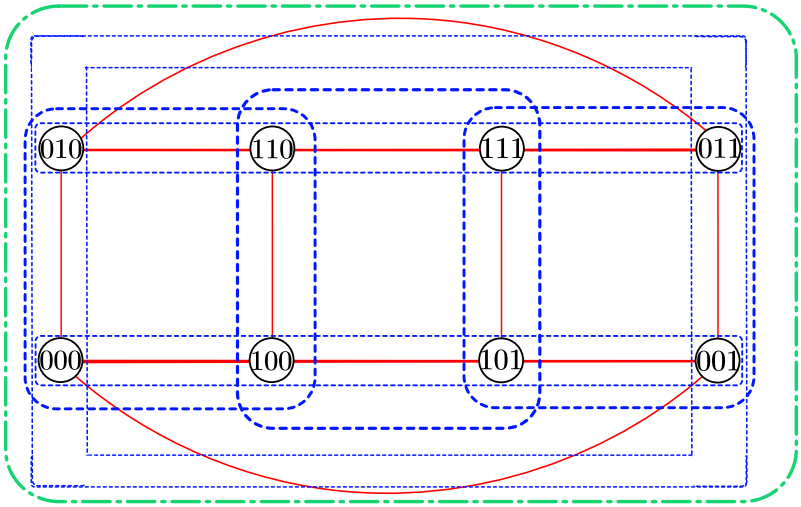


Figure 4.3. Hypergraph $\text{hyp}_{\mathscr{R}_\Omega}$ of the uniform recombination P-structure on three-dimensional binary Hamming sequences $\mathscr{R}_\Omega \stackrel{\text{def}}{=} (\mathcal{H}_2^3, \textsc{Uniform})$. The vertices are illustrated as 'circles', and the hyperedges (i.e. offspring sets of Uniform) are illustrated as 'solid' and 'enclosing lines'. Enclosed vertices define hyperedges; lines shapes/colours are only visual cues. *Red lines*: hyperedges of cardinality two. *Blue dotted lines*: hyperedges of cardinality four. *Green dotted-dashed line*: the hyperedge that includes all vertices.

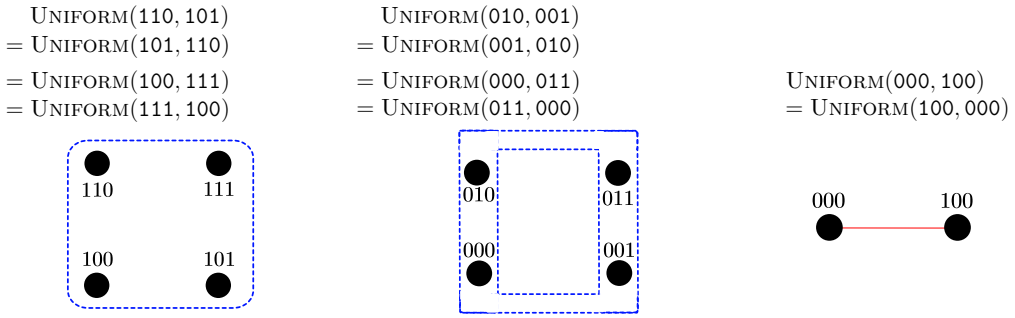


Figure 4.4. Specific hyperedges in the hypergraph $\text{hyp}_{\mathscr{R}_\Omega}$ of the uniform recombination P-structure on three-dimensional binary Hamming sequences $\mathscr{R}_\Omega \stackrel{\text{def}}{=} (\mathcal{H}_2^3, \textsc{Uniform})$, alongside all Uniform recombinations that produce them.

Generally, it is difficult to visualise hypergraphs of recombination P-structures due to the many hyperedges they may contain (one per every pair of parents). One of the least intricate non-trivial examples[4] is the hypergraph $\text{hyp}_{\mathscr{R}_\Omega}$ (Figure 4.3) of the uniform recombination P-structure $\mathscr{R}_\Omega$ seen in Example 4.1. The vertex set of $\text{hyp}_{\mathscr{R}_\Omega}$ consists of all three-dimensional binary Hamming sequences. Its hyperedge

[4]Figure 4.3 omits the incidence matrix $\mathbf{H}$ of the uniform recombination P-structure hypergraph due to its large dimensions. It contains $2^3 \cdot 2^3 \cdot 2^3 = 512$ entries $h_{x,(y,z)}$; $2^3 = 8$ rows for all $x \in \{0,1\}^3$ and $2^3 \cdot 2^3 = 64$ columns for all pairs $(y,z) \in \{0,1\}^3 \times \{0,1\}^3$. Note also that for symmetry of the uniform recombination P-structure, its hypergraph will contain duplicate hyperedges, one for $\textsc{Uniform}(x,y)$ another for $\textsc{Uniform}(y,x)$, even if these offspring sets are equal. Figure 4.3 omits duplicate hyperedges for clarity and displays only one of $\textsc{Uniform}(x,y)$ or $\textsc{Uniform}(y,x)$.

set consists of all offspring sets $\textsc{Uniform}(x, y)$ for all vertices x, y; Figure 4.4 displays some of the hyperedges separately for extra clarity. Each hyperedge is an offspring set, so the vertices connected by a hyperedge are the offspring in an offspring set. For instance, $\{\texttt{000}, \texttt{001}\}$ is a hyperedge connecting only $\texttt{000}$ and $\texttt{001}$ precisely because $\textsc{Uniform}(\texttt{000}, \texttt{001}) = \{\texttt{000}, \texttt{001}\}$; whereas $\textsc{Uniform}(\texttt{000}, \texttt{111}) = \{\texttt{0}, \texttt{1}\}^3$, so $\{\texttt{0}, \texttt{1}\}^3$ is a hyperedge connecting all vertices.

One relevant aspect of hypergraphs is that with hyperedges one can see why candidate solutions may appear closer in recombination search spaces than in mutation search spaces, if 'nearness' is understood as being connected by a hyperedge (or an edge). For example, consider again the uniform recombination P-structure hypergraph $\text{hyp}_{\mathscr{R}_\Omega}$, and recall the hypercube graph in Figure 4.1a representing the mutation search space associated with single-bit flip mutation seen in Section 4.1. Observe that $\{\texttt{000}, \texttt{111}\}$ is not an edge of the hypercube since $\texttt{111}$ cannot be reached in one bit flip from $\texttt{000}$. However, $\texttt{000}$ and $\texttt{111}$ are connected in the hypergraph $\text{hyp}_{\mathscr{R}_\Omega}$ by the hyperedge $\{\texttt{0}, \texttt{1}\}^3$ ('green line' in Figure 4.3). Hence $\texttt{000}$ and $\texttt{111}$ can be said to be closer in the search space associated with uniform recombination than in the search space associated with single-bit flip mutation. Another example would be individuals $\texttt{000}$ and $\texttt{011}$ connected by a hyperedge in $\text{hyp}_{\mathscr{R}_\Omega}$ (Figure 4.4) yet disconnected in the search space associated with single-bit flip mutation since $\{\texttt{000}, \texttt{011}\}$ is not an edge of the hypercube graph.

Hypergraphs of recombination P-structures contain a kind of graphs named *backbone graphs* or *underlying graphs*[5]. These are the subgraph extracted from hypergraphs when selecting only the hyperedges that contain exactly two vertices (i.e. offspring sets of cardinality two).

Definition 4.7 (**Recomb. P-structure backbone graph** [18, 139])**.** Denote $\mathscr{R}$ a recombination P-structure $(X, \mathcal{R})$. Its *backbone graph* is $\text{bbg}_{\mathscr{R}} \stackrel{\text{def}}{=} \big(V(\text{bbg}_{\mathscr{R}}), E(\text{bbg}_{\mathscr{R}})\big)$ where $V(\text{bbg}_{\mathscr{R}}) = X$ is the vertex set, and the edge set is $E(\text{bbg}_{\mathscr{R}}) = \Big\{\{x, y\} \mid x, y \in V(\text{bbg}_{\mathscr{R}}) \text{ and } x \neq y \text{ and } \mathcal{R}(x, y) = \{x, y\}\Big\}$.

Building on earlier work of Culberson [26] and Jones [73], Gitchoff and Wagner [51] show that sometimes the graphs of mutation spaces are precisely the backbone graphs embedded in the hypergraphs of recombination P-structures. This result is known as the *mutation-recombination homomorphism* [51, 139] because the search space structure induced by mutation operators would be preserved by recombination operators.

[5]This thesis may use the term 'underlying graph' for graphs associated with search spaces in general, whether speaking of GF or ELT. The term 'backbone graph' is preferred to avoid confusion.

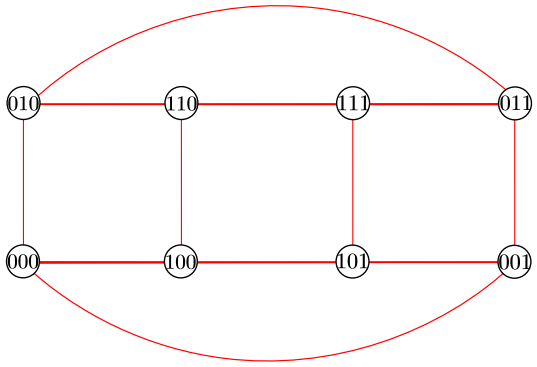


Figure 4.5. The backbone graph, isomorphic to a hypercube graph, embedded in the hypergraph $\text{hyp}_{\mathscr{R}_\Omega}$ of the uniform recombination P-structure on three-dimensional binary Hamming sequences $\mathscr{R}_\Omega \stackrel{\text{def}}{=} (\mathcal{H}_2^3, \textsc{Uniform})$.

A trivial example is given by the three-bit flip neighbourhood $\textsc{Bitflip}(3)$ and the identity recombination P-structure $(\{0,1\}^3, \textsc{Id})$. Here the hypergraph of $(\{0,1\}^3, \textsc{Id})$ is indeed identical (up to isomorphism) to the complete graph K_8 seen earlier in Figure 4.1b. That is, K_8 is the backbone graph for $(\{0,1\}^3, \textsc{Id})$; the reason being that there are no non-trivial hyperedges (i.e. joining more than two vertices) since $\textsc{Id}(x,y) = \{x,y\}$, $\forall x,y \in \{0,1\}^3$. Moreover, the following relationship holds: $\textsc{Id}(x,y) = \{x,y\}$ is an edge of $K_8 \iff y \in \textsc{Bitflip}(3)(x)$ and $x \in \textsc{Bitflip}(3)(y)$.

A non-trivial example is the hypercube associated with the single-bit flip mutation neighbourhood (Figure 4.1a), which is the backbone graph (Figure 4.5) *embedded* in the hypergraph of the uniform recombination P-structure (Figure 4.3). In other words, there is a *structure-preserving map* (i.e. homomorphism) from the mutation search space associated with single-bit flip neighbourhood to the uniform recombination P-structure hypergraph: every edge in mutation space corresponds to a hyperedge in recombination space. For instance, $\{\texttt{000}, \texttt{001}\}$ is an edge of the hypercube in Figure 4.1a and also a hyperedge of the hypergraph in Figure 4.3. But not all hyperedges correspond to ordinary edges in mutation space, so the map is not an isomorphism. A key conclusion of this result is that even when mutation and recombination are defined differently, behave differently and the performance of EAs using them is possibly different, their underlying search space structure may be homomorphic [51, 139]. Part III discusses further details of the mutation-recombination homomorphism and its implications for fitness landscape analysis.

Finally, despite the resemblances between graphs and hypergraphs, there is one important distinction between them when comparing mutation and recombination: from a given graph its associated neighbourhood function can be recovered, but from a given hypergraph its associated recombination P-structure cannot be recov-

ered [51, 139]. To illustrate it with an example, recall the hypercube graph associated with single-bit flip mutation in Figure 4.1a and the hypergraph associated with the uniform recombination P-structure $\mathscr{R}_\Omega$ in Figure 4.3. Unlike the hypercube, where the adjacency structure makes clear which individual is obtained after mutating a given one, from the hypergraph alone one cannot tell which offspring set came from recombining which pair of parents. Hyperedges themselves do not distinguish between offspring and parents, unless $\mathscr{R}_\Omega$ is associated with them. In short, recombination P-structures provide the mappings from parents to offspring absent in hypergraphs, and thus remedy the insufficiency of hypergraphs to describe recombination search spaces.

4.2.3 A Note on Some Similarities and Differences between Recombination P-structures and Geometric Crossovers

This thesis studies side by side recombination P-structures (Definition 4.3) and geometric crossovers (Definition 3.3). To avoid possible confusions, the following points highlight some subtle similarities and differences between them regarding how ELT and GF understand recombination search spaces:

- GF defines search spaces as a metric spaces (X, d) such that geometric mutation and geometric crossover are defined implicitly via a metric d on the domain X, so GF does not distinguish between mutation and recombination search spaces (Chapter 3). By contrast, ELT separates mutation (X, N) and $(X, \mathcal{R})$ recombination search spaces, making explicit that the former use neighbourhoods N and the latter interval operators $\mathcal{R}$ to impose a structure on the domain X (Sections 4.1–4.2). This leads to opposite views on search spaces: in GF search spaces induce search operators, in ELT search operators induce search spaces (i.e. the 'one operator one landscape' view [73]).

- Both geometric crossovers and recombination P-structures rely on the notion of interval to formalise the support functions of recombination operators, but not the same kind of interval. Geometric crossovers use geodesic intervals, so offspring only occur in geodesics between parents. Recombination P-structures instead use intervals that do not require a distance, hence more general than geodesic intervals. In practice, many crossovers are geometric crossovers [100]; however, there exist crossovers in non-metric spaces, and so recombination P-structures are an advantage in such cases. For instance, the unequal crossover on strings is an (infinite) interval operator associated with a non-metrisable

space [131]; realistic examples of non-metrisable spaces can be found in foldings of ribonucleic acid structures [134].

- Hyperedges of hypergraphs represent the support sets of recombination P-structures (Section 4.2.2) but also of geometric crossovers in finite domains at least. For example, recall the hypergraph of the uniform recombination P-structure $(\mathcal{H}_q^n, \textsc{Uniform})$ in Figures 4.3–4.4. Note the hyperedges or offspring sets $\textsc{Uniform}(x, y)$ agree with Hamming metric segments $[x, y]_{d_H}$ or equally finite-complete geometric crossovers (Definition 3.3) under Hamming distance. For instance, $\textsc{Uniform}(\mathtt{011}, \mathtt{000}) = \{\mathtt{011}, \mathtt{001}, \mathtt{010}, \mathtt{000}\} = [\mathtt{011}, \mathtt{000}]_{d_H}$. That is due to Corollary 3.4 in [18]: $\textsc{Uniform}(x, y) = [x, y]_{d_H}$ for all $x, y \in \mathcal{H}_q^n$.

4.2.4 Special Recombination P-structures

Alternative definitions and characterisations of recombination P-structures are possible by selecting, restricting or relaxing their axioms. Two general subclasses of recombination P-structures are *geometric* and *monotonic* recombination P-structures (Definitions 4.8–4.9). Both are introduced in [18] as transit functions using different terminology, which generalise recombination P-structures dropping the size-monotonicity axiom, so Definitions 4.8–4.9 are a particular case of those in [18].

A *geometric recombination P-structure* is a recombination P-structure $(X, \mathcal{R})$ that requires a given 'reference' graph G with vertex set X and one additional axiom: offspring sets $\mathcal{R}(x, y)$ must lie in shortest-paths between x and y on the reference graph G. Hence geometric recombination P-structures are specific cases of geometric crossovers (Definition 3.3) [18]; provided that graphs are instances of metric spaces, and shortest-paths on graphs are instances of metric segments.

Definition 4.8 (**Geometric recomb. P-structure** [18])**.** Let $(X, \mathcal{R})$ be a recombination P-structure and G a connected graph with vertex set X. If $\forall x, y \in X : \mathcal{R}(x, y) \subseteq I(x, y)$, where $I(x, y)$ is the set of all shortest paths between x and y in G, then $(X, \mathcal{R})$ is a *geometric recombination P-structure*. The *subclass* of geometric recombination P-structures is denoted $\mathcal{RP}$-geometric. The *subclass* of *non-geometric* recombination P-structures is: co-$\mathcal{RP}$-geometric $\stackrel{\text{def}}{=} \mathcal{RP} \setminus \mathcal{RP}$-geometric.

For example, the uniform $(\mathcal{H}_q^n, \textsc{Uniform})$ and one-point $(\mathcal{H}_q^n, \textsc{OnePoint})$ recombination P-structures on general Hamming sequences (Examples 4.1–4.2) are geometric recombination P-structures. That is because both uniform and one-point crossovers return offspring in the Hamming metric segment between parents (i.e. in shortest-paths on the Hamming graph) [18].

A salient feature of geometric recombination P-structures, true by definition, is fulfilling the inbreeding properties of geometric crossovers (Section 3.5). In fact, any property true of geometric crossovers that depends solely on offspring being in shortest-paths is true of geometric recombination P-structures, again by definition. Chapter 5 proves that there exists another subclass of recombination P-structures not invoking the definition of geometric crossovers to fulfil the inbreeding properties.

Monotonic recombination P-structures are another relevant subclass of recombination P-structures. They use a stronger version of the size-monotonicity axiom of recombination P-structures: for any children $u, v \in \mathcal{R}(x, y)$ of any parents x and y, the grandchildren $\mathcal{R}(u, v)$ are always a subset of $\mathcal{R}(x, y)$. Size-monotonicity is weaker because grandchildren need not be subsets of children, only that their number must not exceed that of children.

Definition 4.9 (**Monotonic recomb. P-structure** [18, 107])**.** Let $(X, \mathcal{R})$ be a recombination P-structure. Then, $(X, \mathcal{R})$ is a *monotonic recombination P-structure* if $\forall x, y \in X$ and $\forall u, v \in \mathcal{R}(x, y) : \mathcal{R}(u, v) \subseteq \mathcal{R}(x, y)$. The *subclass* of monotonic recombination P-structures is denoted $\mathcal{RP}$-monotonic. The *subclass* of *non-monotonic* recombination P-structures is: co-$\mathcal{RP}$-monotonic $\stackrel{\text{def}}{=} \mathcal{RP} \setminus \mathcal{RP}$-monotonic.

Uniform recombination P-structures (Example 4.1) are monotonic. For instance, given parents `010` and `001`

$$\textsc{Uniform}(\overset{u}{\overbrace{\texttt{010}}}, \overset{v}{\overbrace{\texttt{011}}}) = \{\texttt{010}, \texttt{011}\} \subset \{\texttt{000}, \texttt{001}, \texttt{010}, \texttt{011}\} = \textsc{Uniform}(\overset{x}{\overbrace{\texttt{010}}}, \overset{y}{\overbrace{\texttt{001}}}) \ . \tag{4.2}$$

Other multi-point recombination P-structures such as one-point (Example 4.2) are not monotonic in general [18]. For example, given parents `000` and `111`

$$\underbrace{\textsc{OnePoint}(\overset{u}{\overbrace{\texttt{100}}}, \overset{v}{\overbrace{\texttt{011}}})}_{\{\texttt{010,101}\}\text{ member}} \not\subseteq \underbrace{\textsc{OnePoint}(\overset{x}{\overbrace{\texttt{000}}}, \overset{y}{\overbrace{\texttt{111}}})}_{\{\texttt{010,101}\}\text{ not a member}} \ . \tag{4.3}$$

Chapter 5

A Classification of Crossovers

This chapter is an original major contribution of this thesis, extending two papers that I co-authored with Moraglio [48, 49]. This chapter presents a formal classification of crossovers based on the classes of geometric crossovers and recombination P-structures, reviewed in Chapters 3 and 4 respectively, independent of specific problems and representation of solutions.

5.1 Introduction

Classification is a systematic process of organising entities into groups called *classes*, providing a rigorous means for the accumulation, retrieval and communication of knowledge about such entities. It has been a fundamental tool across many disciplines, and evolutionary computation is no exception [141].

This chapter presents a classification of crossovers, given by their support functions, with respect to the classes defined by geometric crossovers (Chapter 3) and recombination P-structures (Chapter 4). That is, a hierarchy of crossover classes independent of problems and representations that classifies crossovers in terms of their parent-offspring reachability structure, transparent to their probabilistic nature, and constitutes the foundations to unify the GF and ELT.

In the GF and ELT the reachability structure of crossovers, within an ambient search space, plays a key role in crossover design and EAs' behaviour and performance. For example, the GF uses distances associated with search spaces to design specific crossovers across representations as well as to understand why and how geometric-crossover EAs, polynomial performance and abstract convex landscapes relate to each other [100, 104]. ELT relies on the reachability structure of mutation and crossover operators to compare operator performance via landscape correlation

and epistatic interactions in the genotype of solutions, corroborate Holland's schema theorem [65], or explain why commutativity of mutation and crossovers leads to unbiasedness in evolutionary search [136, 139, 140]. Despite the GF and ELT adopt a general, representation-problem independent approach to study crossovers, their views remain incomplete.

The GF regards the search space as a set of solutions equipped with a metric, to formalise crossovers across representations through geodesic intervals (Section 3.2); however, this approach is inherently limited to search spaces where distances must be defined, and little is known beyond a coarse classification of crossovers into geometric and non-geometric (Proposition 3.1). In contrast, ELT allows more general (i.e. less structured) search spaces by defining them using sets not metrics, leading to recombination P-structures as a specific case of general interval functions to formalise crossover (Section 4.2), but ELT never addressed their classification.

This chapter aims to remediate such partial views on crossover with the proposed crossover classification, thereby setting a basis to understand how the structure of crossovers may affect their design or the behaviour and performance of EAs using them. This chapter will extend the classification of geometric versus non-geometric crossovers in the GF (Chapter 3) and propose one for the first time in ELT. Its contributions are:

1. A formal justification that it is theoretically possible and meaningful to compare geometric crossovers and recombination P-structures to develop a classification of crossovers in the GF and ELT. (Section 5.3.)

2. A crossover classification with respect to geometric crossovers and recombination P-structures. Section 5.2 overviews it and Section 5.4 develops it.

3. Two case studies to demonstrate how the crossover classification may be used. The first checks the existence of a crossover in the classification with a specific property: high-locality [144]. The second defines a new subclass of recombination P-structures fulfilling the inbreeding properties of geometric crossovers [103]. (Section 5.5.)

4. Proving that high-locality [144] and geometricity of crossovers [100] are not always equivalent concepts. (Section 5.5.1.)

5. Providing evidence to support the conjecture that the inbreeding properties proposed in [103] (see Chapter 3) are insufficient to determine if a crossover is geometric and thus can only be used to test non-geometricity. (Section 5.5.2.)

Section 5.6 concludes this chapter.

5.2 Overview and Methodology

This section overviews this chapter by summarising what crossovers and (sub)classes are considered in the classification, an illustration of such classification, and outlining in Section 5.2.1 three approaches to exploit it.

The crossover classification consists of the following major classes and subclasses of geometric crossovers and recombination P-structures outlined in Table 5.1, all of which were introduced in Chapters 3–4.

No.	Name	Notation	Class/Subclass	Definition
1	Universal	$\mathcal{U}$	Class	All crossovers
2	Geometric crossovers	$\mathcal{GX}$	Class	3.3
3	Recombination P-structures	$\mathcal{RP}$	Class	4.3
4	Non-geometric crossovers	$\overline{\mathcal{GX}}$	Class	$(\mathcal{U} \setminus \mathcal{GX})$
5	Not recombination P-structures	$\overline{\mathcal{RP}}$	Class	$(\mathcal{U} \setminus \mathcal{RP})$
6	Finite-complete geometric crossovers	$\mathcal{GX}$-complete$_{fin}$	Subclass	3.3
7	Geometric recombination P-structures	$\mathcal{RP}$-geometric	Subclass	4.8
8	Monotonic recombination P-structures	$\mathcal{RP}$-monotonic	Subclass	4.9
9	Finite-incomplete geometric crossovers	co-$\mathcal{GX}$-complete$_{fin}$	Subclass	3.3
10	Non-geometric recombination P-structures	co-$\mathcal{RP}$-geometric	Subclass	$(\mathcal{RP} \setminus \mathcal{RP}$-geometric$)$
11	Non-monotonic recombination P-structures	co-$\mathcal{RP}$-monotonic	Subclass	$(\mathcal{RP} \setminus \mathcal{RP}$-monotonic$)$

Table 5.1. Crossover (sub)classes in the crossover classification.

The choice of geometric crossovers and recombination P-structure classes is evident since they are an integral part of the GF and ELT, and this chapter aims to set the foundations of a unified theory upon such classes. The subclasses (6–8), or their complements (9–11), in Table 5.1 are chosen for their salient features:

- Complete geometric crossovers coincide with geodesic intervals: the offspring

are all the points in shortest-distance trajectories between the endpoints of a geodesic interval. This makes complete geometric crossovers enjoy the properties of geodesic intervals [146] (e.g. symmetry) that other geometric crossovers may not have.

- Geometric recombination P-structures, being a special case of geometric crossovers, fulfil the inbreeding properties (Propositions 3.2–3.4). Hence the inbreeding properties are always present in geometric recombination P-structures, as they are for geometric crossovers, across representations and problems.

- Monotonic recombination P-structures have the particular property that offspring sets are always closed or invariant by Definition 4.9. That is, all possible descendants of two parents x and y are confined to the offspring set $\mathcal{R}(x, y)$, so $\mathcal{R}(u, v) \subseteq \mathcal{R}(x, y)$ for all $u, v \in \mathcal{R}(x, y)$. A similar notion of invariance was used in [98, 99, 117] to generalise Holland's schema [65].

Crossover examples of these (sub)classes were seen in Chapters 3–4 as well, but this chapter requires additional ones to build the classification, so this chapter recalls or introduces them as needed. Table 5.2 summarises key details about all the crossovers to be classified.

Name	Notation	Search space	Definition
Uniform (unbiased)	UNIFORM	Hamming sequences	3.5
Asymmetric one-point	asym-ONEPOINT	Hamming sequences	3.6
One-point	ONEPOINT	Hamming sequences	4.5
Koza's subtree swap	KOZA	Labelled trees in GP	5.1
Davis's order	DAVIS	Permutations	5.2
Symmetric Davis's order	sym-DAVIS	Permutations	5.3
All-Hamming-paths	ALLPATHS($H(n, q)$)	Hamming sequences	5.4
Intersecting-Hamming-balls	BALLS(d_{H})	Hamming sequences	5.6
Identity	ID	Generic: finite spaces	4.4

Table 5.2. Crossover support functions in the crossover classification.

The classification that results from these crossovers and (sub)classes is illustrated in the following Figure 5.1. The universal class of crossovers $\mathcal{U}$ is at the top of the hierarchy, containing the two fundamental classes: geometric crossovers $\mathcal{GX}$ and recombination P-structures $\mathcal{RP}$. Within their intersection, geometric recombination P-structures $\mathcal{RP}$-geometric form a superclass of finite-complete geometric crossovers $\mathcal{GX}$-complete$_{fin}$. By contrast, the monotonic recombination P-structures subclass

$\mathcal{RP}$-monotonic is not completely within the intersection of $\mathcal{GX}$ and $\mathcal{RP}$ (see all-Hamming-paths crossover), but $\mathcal{RP}$-monotonic does share crossovers with both of $\mathcal{GX}$-complete$_{fin}$ and $\mathcal{RP}$-geometric (e.g uniform and identity crossovers). Finally, neither $\mathcal{GX}$ nor $\mathcal{RP}$ cover all existing crossovers in $\mathcal{U}$ (e.g. Koza's subtree swap, Davis's order crossover and its symmetric version).

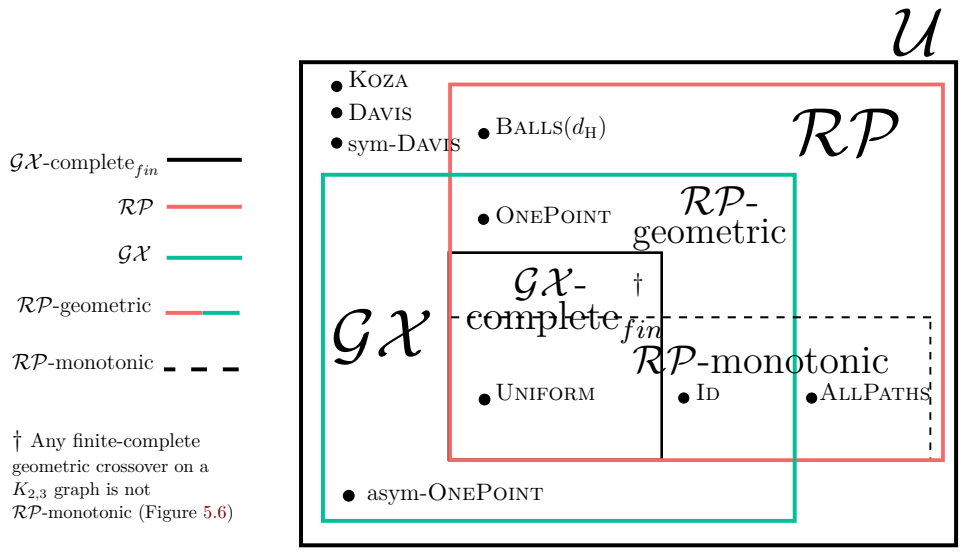


Figure 5.1. Classification within the universal class $\mathcal{U}$ of crossovers by (sub)classes of geometric crossovers and recombination P-structures. *Classes*: geometric crossovers $\mathcal{GX}$ and recombination P-structures $\mathcal{RP}$. *Subclasses*: finite-complete geometric crossovers $\mathcal{GX}$-complete$_{fin}$, geometric recombination P-structures $\mathcal{RP}$-geometric and monotonic recombination P-structures $\mathcal{RP}$-monotonic. *Crossover examples*: (unbiased) uniform (Uniform), one-point (OnePoint), asymmetric one-point (asym-OnePoint), Koza's subtree swap (Koza), Davis's order (Davis), symmetric Davis's order (sym-Davis), all-Hamming-paths (AllPaths($H(n,q)$)), intersecting-Hamming-balls (Balls(d_H)) and identity (Id).

By its very nature, this classification does not involve other aspects of crossovers than their structure. For example, probability distributions over offspring, algorithmic implementation details, or practicality on a certain problem have been left out. Not because they are unimportant but because each of them deserves a separate study. Nevertheless, from the discussion in Section 5.4, those other aspects are not completely unrelated to this classification inasmuch as the structure of crossovers plays a key role in them. The rest of this chapter consists in justifying why a unified theory of the GF and ELT grounded on this classification is not futile (Section 5.3), proving all necessary results that lead to this classification (Section 5.4), and then two case studies to demonstrate its utility (Section 5.5).

5.2.1 Methodology

Three possible general strategies to use the crossover classification are outlined next: bottom-up, top-down and inter-class. Although these are not the main focus of this chapter, a brief explanation can help appreciate better the overall utility of the classification and how the two case studies in Section 5.5 use it in particular.

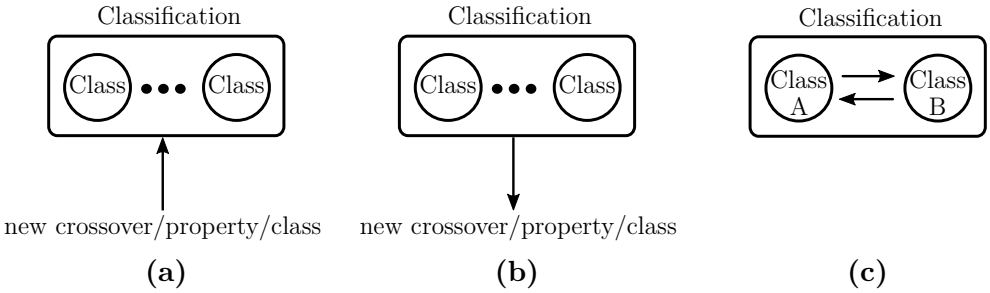


Figure 5.2. Three strategies to use the classification of crossovers: bottom-up (Figure 5.2a), top-down (Figure 5.2b) and inter-class (Figure 5.2c).

The *bottom-up* strategy (Figure 5.2a) is about classifying and generalising previously unclassified crossovers, classes of crossovers or specific properties. For example, given a crossover or a class, one proves which classes in the hierarchy include the given crossover or class. In this manner, one not just finds the associated structure (whether geometric or not) but at the same time obtains the superclasses that generalise the given crossover or class. Likewise, for a given property, one proves whether there exist a crossover or class in the hierarchy fulfilling such property. Sections 5.4 and 5.5.1 follow this strategy.

The *top-down* strategy (Figure 5.2b) is about specialisation of classes and design of specific crossovers from those already present in the hierarchy, rather than generalisation as in bottom-up. Here, one starts with a given class in the hierarchy and from its axioms proposes specific crossovers that belong to such class. Or, modifying the axioms of the given class (i.e. adding, removing, restricting or relaxing them), one derives more specific classes with desired properties. For example, geometric and monotonic recombination P-structures are obtained from recombination P-structures by requiring one additional axiom for each of them (Section 4.2.4). Section 5.5.2 follows this strategy.

Finally, the *inter-class* strategy (Figure 5.2c) is about comparing two or more classes in the hierarchy through their axioms or definitions. For example, given two seemingly disjoint classes, one may want to prove if they are actually disjoint. If one of the classes verifies the axioms of the other class, then the former is included in the latter, and so the classes were not disjoint after all. Indeed, this is precisely what

Theorem 5.1 does in Section 5.3 to show that finite-complete geometric crossovers are a subclass of recombination P-structures. This strategy is useful even if the classes are already present in the classification because not all their relationships may be known a priori. Doing a class-to-class comparison helps us find new relationships or clarify existing ones.

5.3 Crossover Classification is Possible

Geometric crossovers $\mathcal{GX}$ and recombination P-structures $\mathcal{RP}$ originate in separate theories with distinct aims and views on search spaces (Section 5.1), and a first inspection of their respective Definitions 3.3 and 4.3 reveals little about their relationship. Are these two classes of crossovers related in any meaningful way or are they disjoint? If $\mathcal{GX}$ and $\mathcal{RP}$ were disjoint classes, then no axiomatic relationships other than disjointness could be established between geometric crossovers and recombination P-structures since their axioms would represent distinct crossovers having nothing in common. Consequently, developing a hierarchy of subclasses of $\mathcal{GX}$ and $\mathcal{RP}$ and comparing crossovers based on such hierarchy would be futile. Knowing whether $\mathcal{GX}$ and $\mathcal{RP}$ are disjoint (Question 5.1) is fundamental to ensure that unifying the GF and ELT is not futile.

Question 5.1. *Are the classes defined by geometric crossovers $\mathcal{GX}$ and recombination P-structures $\mathcal{RP}$ disjoint?*

Theorem 5.1 proves that the answer to Question 5.1 is: no. Geometric crossovers and recombination P-structures are not disjoint because finite-complete geometric crossovers $\mathcal{GX}$-complete$_{fin}$ (Definition 3.3) are a subclass of geometric crossovers $\mathcal{GX}$ (Proposition 3.1) but also a subclass of recombination P-structures $\mathcal{RP}$ (Theorem 5.1). Hence Theorem 5.1 justifies that pursuing a unified theory of geometric crossovers and recombination P-structures remains possible.

Theorem 5.1 ($\mathcal{GX}$-complete$_{fin}$ crossovers are recomb. P-structures [48]). Let (X, d) be any metric space with a finite set X and any finite-complete geometric crossover $\xi(x, y) = [x, y]_d$ for arbitrary parents $x, y \in X$. Then, (X, ξ) is a recombination P-structure.

Proof. Let us prove that metric segments fulfil the axioms of recombination P-structures.

(I) *Fix-point.* The only possible z in $[x, x]_d = \{z \in X \mid d(x, z) + d(z, x) = d(x, x)\}$ is exactly x. Therefore, $[x, x]_d = \{x\}$.

(II) *Symmetry.* $[x, y]_d = [y, x]_d$ follows immediately from the symmetry axiom of metric spaces (Definition 3.1).

(III) *Null-recombination.* $\{x, y\} \subseteq [x, y]_d$ holds by definition because in a closed metric segment the extremes, x and y, are always included.

(IV) *Size-monotonicity.* To prove that if $z \in [x, y]_d$, then $\left|[x, z]_d\right| \leq \left|[x, y]_d\right|$, it suffices to know that metric segments fulfil monotonicity (see Proposition 4.6.1 in [146]): $\forall x, y, z \in X$ if $z \in [x, y]_d$ then $[x, z]_d \subseteq [x, y]_d$. Therefore, it follows that $\left|[x, z]_d\right| \leq \left|[x, y]_d\right|$. □

Geodesic intervals are a well known type of interval operators enjoying a rich set of properties [107, 112, 146], some of them stronger than those fulfilled by recombination P-structures (e.g. monotonicity in Proposition 4.6.1 [146] compared with size-monotonicity), and so Theorem 5.1 may appear unsurprising. But Theorem 5.1 is significant because it makes possible to unify the GF and ELT based on a classification of crossovers. That is, since the geometric crossovers and recombination P-structures classes are not disjoint, it is possible and worth knowing to what extent crossovers in both classes share common properties, as the remainder of this chapter elaborates. Theorem 5.1 makes explicit that geometric crossovers and recombination P-structures have more in common than what appears at first glance.

Recall from Chapter 3 that Proposition 3.1, proving that not all crossovers are geometric, is essential for the GF to guarantee its viability as a unified theory of EAs. Theorem 5.1 shows that finite-complete geometric crossovers are a subclass of recombination P-structures, but it leaves open another fundamental question to ensure that unifying the GF and EL is not futile.

Question 5.2. *Does the recombination P-structures $\mathcal{RP}$ class include all crossovers?*

Theorem 5.2 proves that the answer to Question 5.2 is: no. Not all crossovers are recombination P-structures, so they split into the class $\mathcal{RP}$ and its complement $\overline{\mathcal{RP}}$ with respect to the universal class of all crossovers.

Theorem 5.2 (**Existence of $\overline{\mathcal{RP}}$** [49])**.** The class of crossovers that are not recombination P-structures $\overline{\mathcal{RP}}$ is not empty. Therefore, crossover operators split into two classes: recombination P-structures $\mathcal{RP}$ and not recombination P-structures $\overline{\mathcal{RP}}$.

Proof. The asymmetric one-point crossover (Definition 3.6) is not a recombination P-structure since it violates the symmetry axiom of recombination P-structures (Remark 4.2). Other examples are crossovers that never return the parents in the

offspring set, thus violating the null-recombination axiom; for example geometric crossovers ξ (Definition 3.3) whose offspring sets coincide with *open* geodesic intervals, that is $\xi(x, y) = [x, y]_d \setminus \{x, y\}$ for some parents x, y and metric d. □

Theorem 5.2 proves the analogous of Proposition 3.1 for recombination P-structures, and together they ensure that a unified theory of geometric crossovers and recombination P-structures is not a theory of all crossovers. This is important for the viability of unifying the GF and EL for essentially the same reasons Proposition 3.1 is vital to the GF (Section 3.4). First, a theory of all crossovers can be a symptom of an overgeneralised theory that leads to meaningless or tautological statements about crossovers, so in this sense the limitation imposed by Theorem 5.2 over recombination P-structures, and that of Proposition 3.1 over geometric crossovers, is more of an advantage than a disadvantage. Secondly, embedding problem knowledge (i.e. landscape features) in an EA effectively entails restricting the search operators it uses, so that the EA is tailored to the problem (or a class thereof) it wants to solve. Therefore, a theory of all crossovers where no restrictions are imposed would be futile in principle, suggesting that a meaningful theory that matches EAs and problems cannot involve all possible crossover operators. Finally, although examples of crossovers that are not recombination P-structures are known in the literature (e.g. unequal crossover [131]), the classes $\mathcal{RP}$ and $\overline{\mathcal{RP}}$ were not defined explicitly. Theorem 5.2 makes clear the existence of these classes and proves it using other examples than unequal crossover, namely the asymmetric one-point crossover and every crossover whose offspring sets always exclude parents.

In contrast with geometric crossovers, what is still unclear is the practicality of recombination P-structures. That is, whether they include many crossovers used in practice; hardly any concrete examples are known besides multi-point crossovers on strings [51, 139]. Nevertheless, recombination P-structures are promising, for they overlap with geometric crossovers (Theorem 5.1), which include many crossovers used in practice across solution representations and problems [100]. Section 5.4 contributes in that regard by presenting other examples of recombination P-structures and a classification that can help us find new ones.

5.4 A Classification of Crossovers

This section develops a classification of crossovers upon the (sub)classes of geometric crossovers and recombination P-structures overviewed in Section 5.2. First, Sections 5.4.1–5.4.5 identify the corresponding classes in Table 5.2 for each crossover

in Table 5.1. Then, Section 5.4.6 presents the main result in Theorem 5.15, observed earlier in Figure 5.1, establishing the key relationships between the (sub)classes of geometric crossovers and recombination P-structures that constitute the classification. Contributions overall are discussed at the end of Section 5.4.6.

5.4.1 Uniform and (Asymmetric-)One-point

The classes to which uniform and (asymmetric-)one-point crossovers belong are already known from Chapters 3–4 and from [18, 100]. Next, Theorems 5.3–5.5 make it explicit for the classification of crossovers.

Theorem 5.3 (UNIFORM $\in \mathcal{GX}$-complete$_{fin}$, $\mathcal{RP}$-geometric, $\mathcal{RP}$-monotonic)**.** Let $(\mathcal{H}_q^n, d_{\mathrm{H}})$ be any n-dimensional q-ary ($q \geq 2$) Hamming metric space. Then, the (unbiased) uniform crossover UNIFORM is a finite-complete geometric crossover on $(\mathcal{H}_q^n, d_{\mathrm{H}})$. Also, $(\mathcal{H}_q^n, \textsc{Uniform})$ is both a geometric and monotonic recombination P-structure.

Proof. Let us prove first that UNIFORM is a finite-complete geometric crossover, and then that it is both a geometric and monotonic recombination P-structure:

1. UNIFORM is a finite-complete geometric crossover on $(\mathcal{H}_q^n, d_{\mathrm{H}})$ because its offspring sets equal the Hamming metric segment: $\textsc{Uniform}(x, y) = [x, y]_{d_{\mathrm{H}}}$ for all parents $x, y \in \mathcal{H}_q^n$. Theorem 3.4.2 in [100] proved it for binary Hamming sequences $\mathcal{H}_2^n$ and Corollary 3.4 in [18] for general Hamming sequences $\mathcal{H}_q^n$.

2. Because UNIFORM is a finite-complete geometric crossover, $(\mathcal{H}_q^n, \textsc{Uniform})$ is a recombination P-structure by Theorem 5.1, and by Definition 4.8 also a geometric recombination P-structure since UNIFORM outputs offspring in shortest-paths between parents on the Hamming graph induced by $(\mathcal{H}_q^n, d_{\mathrm{H}})$.

3. Corollary 3.4 in [18] proves that $(\mathcal{H}_q^n, \textsc{Uniform})$ is a monotonic recombination P-structure: $\textsc{Uniform}(u, v) \subseteq \textsc{Uniform}(x, y)$ for all $x, y \in \mathcal{H}_q^n$ and for all $u, v \in \textsc{Uniform}(x, y)$. See Equation 4.2 for an example. □

Theorem 5.4 (ONEPOINT $\in$ co-$\mathcal{GX}$-complete$_{fin}$, $\mathcal{RP}$-geometric, co-$\mathcal{RP}$-monotonic)**.** Let $(\mathcal{H}_q^n, d_{\mathrm{H}})$ be any n-dimensional q-ary ($q \geq 2$) Hamming metric space. Then, the one-point crossover ONEPOINT is a finite-incomplete geometric crossover on $(\mathcal{H}_q^n, d_{\mathrm{H}})$. Also, $(\mathcal{H}_q^n, \textsc{OnePoint})$ is a geometric recombination P-structure but not monotonic.

Proof. Let us prove first that OnePoint is a finite-incomplete geometric crossover, and then that it is a geometric recombination P-structure but not monotonic:

1. It follows from Theorem 5.3 that OnePoint is a finite-incomplete geometric crossover because its offspring sets are subsets of Uniform: $\text{OnePoint}(x, y) \subseteq \text{Uniform}(x, y)$ for all parents $x, y \in \mathcal{H}_q^n$ [18].

2. $(\mathcal{H}_q^n, \text{OnePoint})$ is a recombination P-structure [139]. Since OnePoint is a finite-incomplete geometric crossover, it produces offspring in shortest-paths between parents, and therefore $(\mathcal{H}_q^n, \text{OnePoint})$ is also a geometric recombination P-structure by Definition 4.8.

3. In general, one-point crossover does not fulfil Definition 4.9 of monotonic recombination P-structures [18]: $\text{OnePoint}(u, v) \not\subseteq \text{OnePoint}(x, y)$ for all $u, v \in \text{OnePoint}(x, y)$ and all $x, y \in \mathcal{H}_q^n$. See Equation 4.3 for a counterexample. □

Theorem 5.5 (asym-OnePoint $\in$ co-$\mathcal{GX}$-complete$_{fin}$, $\overline{\mathcal{RP}}$)**.** Let $(\mathcal{H}_q^n, d_{\text{H}})$ be any n-dimensional q-ary ($q \geq 2$) Hamming metric space. Then, the asymmetric one-point crossover asym-OnePoint is a finite-incomplete geometric crossover on $(\mathcal{H}_q^n, d_{\text{H}})$. However, $(\mathcal{H}_q^n, \text{asym-OnePoint})$ is not a recombination P-structure.

Proof. Let us prove first that asym-OnePoint is a finite-incomplete geometric crossover, and then that it is not a recombination P-structure:

1. Because the offspring sets of asym-OnePoint are a subset of OnePoint (Definition 4.5), it follows immediately from Theorem 5.4 that asym-OnePoint is a finite-incomplete geometric crossover.

2. $(\mathcal{H}_q^n, \text{asym-OnePoint})$ is not a recombination P-structure because it fails the symmetry axiom (Remark 4.2): there exist parents $x, y \in \mathcal{H}_q^n$ such that $\text{asym-OnePoint}(x, y) \neq \text{asym-OnePoint}(y, x)$. □

One interesting insight from Theorems 5.4–5.5 is that even when two crossovers 'look similar', such as asym-OnePoint and its symmetric version OnePoint, formally their classes can be completely separate. That is, one-point crossover is a recombination P-structure (OnePoint $\in \mathcal{RP}$) but asymmetric one-point crossover is not (asym-OnePoint $\in \overline{\mathcal{RP}}$). Consequently, in cases where the difference is less obvious, having a classification would help to formally distinguish the structure of otherwise 'seemingly identical' crossovers.

5.4.2 Koza's Subtree Swap and Davis's Order

Koza's subtree swap and *Davis's order* (Definitions 5.1–5.2) are two well known crossovers to recombine genetic programs in GP and, respectively, permutations in combinatorial optimisation problems like the TSP [4]. Both are non-geometric crossovers [103], for they fail the purity and convergence inbreeding properties respectively (see Figure 3.2 in Section 3.4 for an example). This section proves that Koza's subtree swap and Davis's order neither are recombination P-structures.

Definition 5.1 (**Koza's subtree swap crossover** [49, 86])**.** Let parents $x, y \in T$, where T is a non-empty set of labelled ordered trees. Select uniformly at random one node from each parent: i from x, and j from y. An offspring pair (s, t) results from swapping the subtrees with root nodes i and j. This crossover, parametrised with the chosen nodes i and j, is denoted $\textsc{Koza}(i, j) : T \times T \to T \times T$, $(x, y) \mapsto (s, t) = \textsc{Koza}(i, j)(x, y)$ for parents x and y with offspring s and t. The *support function* is $\textsc{Koza}(x, y) \stackrel{\text{def}}{=} \bigcup_{i \in V(x),\, j \in V(y)} \textsc{Koza}(i, j)(x, y)$, where $V(x)$ and $V(y)$ are the vertex sets of the labelled ordered trees x, y.

Theorem 5.6 ($\textsc{Koza} \in \overline{\mathcal{RP}}$ [49])**.** Let T be any non-empty finite set of labelled ordered trees. Then, $(T, \textsc{Koza})$ is not a recombination P-structure.

Proof. Koza subtree swap is not a geometric crossover (Theorem 5 in [103]) because recombining two identical parents sometimes results in a different offspring, thus violating the purity inbreeding property (Proposition 3.2). Consequently, it also violates the fix-point axiom of recombination P-structures requiring $\mathcal{R}(x, x) = \{x\}$ for every parent x. For instance, Figure 3.2 shows an example where swapping the subtrees with root nodes $i = 2$ (node 'x') in the first parent and $j = 3$ (node '$*$') in the second identical parent leads to offspring different from parents. □

Theorem 5.6 proves then, alongside [103], Koza's subtree swap crossover is neither a geometric crossover nor a recombination P-structure. Next, Theorem 5.7 proves the analogous result for Davis's order crossover (Definition 5.2) and a new symmetric version (Definition 5.3) that I proposed in [49].

Definition 5.2 (**Davis's order crossover** [31, 49])**.** Let individuals $x, y, z \in \mathcal{S}_X$ be permutations of any non-empty finite set X with n elements. Randomly pick indices i and j delimiting the crossover section[1] such that $1 \leq i \leq j \leq n$. Then from the first parent x (the 'cutter') copy the genes in the section $x_i, \ldots, x_j$ and paste them into

[1] A crossover section with all parent's genes, namely $i = 1$ and $j = n$, is also a valid contiguous section, even if this is not usually done in the literature [4].

the offspring through $z_i, \ldots, z_j$ in the same order. Finally, from left to right, fill in order the remaining positions of z using the genes of the second parent y (the 'filler') except those that have been copied. This crossover, parametrised with crossover section indices i and j, is denoted $\textsc{Davis}(i,j) : \mathcal{S}_X \times \mathcal{S}_X \to \mathcal{S}_X$, $(x,y) \mapsto z = \textsc{Davis}(i,j)(x,y)$. The *support function* is $\textsc{Davis}(x,y) \stackrel{\text{def}}{=} \bigcup_{1 \leq i \leq j \leq n} \textsc{Davis}(i,j)(x,y)$.

Example 5.1 (**Davis's order crossover is asymmetric** [49])**.** Let parents $x = \texttt{312}$ (the 'cutter') and $y = \texttt{123}$ (the 'filler'). Consider offspring $\texttt{2}\underline{\texttt{1}}\texttt{3} = \textsc{Davis}(2,2)(\texttt{3}\underline{\texttt{1}}\texttt{2}, \texttt{123})$, where $\underline{\texttt{1}}$ is the crossover section. Notice $\texttt{213}$ cannot be generated by $\textsc{Davis}(i,j)(\texttt{123}, \texttt{312})$ for any crossover section indices i and j because it is not possible to generate $\texttt{2}$ at the first position. Either the second parent places $\texttt{3}$ or $\texttt{1}$ as fillers, or the first parent places $\texttt{1}$ with the crossover section. Therefore, $\textsc{Davis}(i,j)(\texttt{312}, \texttt{123}) \neq \textsc{Davis}(i,j)(\texttt{123}, \texttt{312})$ for all $1 \leq i \leq j \leq 3$.

Davis's order crossover does not meet the symmetry axiom of recombination P-structures in general (Example 5.1). But its symmetric version (Definition 5.3) achieves it in a similar fashion to how one-point crossover (Definition 4.5) symmetrises the asymmetric one-point crossover (Definition 3.6): using both parents as the 'first' and 'second' parent.

Definition 5.3 (**Symmetric Davis's order crossover** [49])**.** Let any parents $x, y \in \mathcal{S}_X$ be permutations of any non-empty finite set X with n elements. Then, $\text{sym-}\textsc{Davis}(x,y) \stackrel{\text{def}}{=} \bigcup_{1 \leq i \leq j \leq n} \{\textsc{Davis}(i,j)(x,y), \textsc{Davis}(i,j)(y,x)\}$ is the *symmetric Davis's order crossover*, producing all possible offspring of parents x and y over all possible crossover sections.

Example 5.2 (**Symmetric Davis's order crossover: not size-monotonic** [49])**.** Using the same parents of Example 5.1: $\text{sym-}\textsc{Davis}(\texttt{312}, \texttt{123}) = \{\texttt{312}, \texttt{123}, \texttt{321}, \texttt{132}, \texttt{213}\}$. That is, $|\text{sym-}\textsc{Davis}(\texttt{312}, \texttt{123})| = 5$. Recombining now the first parent $\texttt{312}$ with offspring $\texttt{213}$: $\text{sym-}\textsc{Davis}(\texttt{312}, \texttt{213}) = \{\texttt{231}\} \cup \text{sym-}\textsc{Davis}(\texttt{312}, \texttt{123})$. Therefore, $|\text{sym-}\textsc{Davis}(\texttt{312}, \texttt{213})| = 6 > 5 = |\text{sym-}\textsc{Davis}(\texttt{312}, \texttt{123})|$, so sym-Davis is not size-monotonic.

Example 5.2 is one of the simplest counterexamples to disprove size-monotonicity of sym-Davis crossover. To help finding other less trivial counterexamples in an automated way, Appendix A.2 provides a software implementation of sym-Davis.

Remark 5.1. Example 5.2 also shows that symmetric Davis's order crossover is non-geometric since it fails the convergence inbreeding property. For instance, $\texttt{213} \in \text{sym-}\textsc{Davis}(\texttt{312}, \texttt{123})$, and $\texttt{123} \in \text{sym-}\textsc{Davis}(\texttt{312}, \texttt{213})$.

Theorem 5.7 (DAVIS, sym-DAVIS $\in \overline{\mathcal{RP}}$ [49])**.** Let $\mathcal{S}_X$ be permutations of any non-empty finite set X. Then, neither $(\mathcal{S}_X, \text{DAVIS})$ nor $(\mathcal{S}_X, \text{sym-DAVIS})$ are recombination P-structures.

Proof. Examples 5.1–5.2 provide counterexamples respectively. □

5.4.3 All-paths

The *all-paths* transit (or interval) function of a finite, connected, and undirected graph G is a function that returns the set of vertices lying on any path between two given vertices of G [17]. Recall that a *path* is a sequence without repeated vertices, and a *cycle* is a sequence (with at least three vertices) where the first and last vertex repeat [59], so all-paths does not return cycles.

Here, all-paths is deemed a 'generic' crossover by interpreting the given vertices as parents and the output set as their offspring (Definition 5.4). Generic in the sense that the vertex set of G, where all-paths is defined, is not bound to a particular search space other than being a finite set (e.g Hamming sequences, permutations, etc.). Example 5.3 shows a specific case on hypercubes.

Definition 5.4 (**All-paths crossover** [17])**.** Let vertices $x, y, z \in V$ of a 'reference' graph G. Then, the *all-paths crossover* is $\text{ALLPATHS}(G) : V \times V \to \mathcal{P}(V)$, $(x, y) \mapsto \{z \mid z \text{ lies on any } x\text{-}y \text{ path in } G\}$, parametrised with G, returning all offspring z lying on any path between parents x and y. It is denoted ALLPATHS when it is clear from the context which graph it refers to.

Example 5.3 (**All-paths crossover in hypercube graphs** [49])**.** Consider the three dimensional hypercube graph. If the chosen parents are identical, then offspring equal parents: $\text{ALLPATHS}(\texttt{001}, \texttt{001}) = \texttt{001}$. The only valid path from $\texttt{001}$ to itself is the one-element sequence $(\texttt{001})$. For example, $(\texttt{001}, \texttt{000}, \texttt{001})$ is a cycle, not a path, thus $\texttt{000} \notin \text{ALLPATHS}(\texttt{001}, \texttt{001})$. If parents are different, then offspring equal the whole vertex set: $\text{ALLPATHS}(\texttt{000}, \texttt{001}) = \{\texttt{0}, \texttt{1}\}^3$.

Remark 5.2 (To Example 5.3)**.** In general, all-paths does not return all vertices of the given graph for different parents, such as in star graphs or trees for example. Hamming graphs are an exception (Lemma 5.1).

This section proves that all-paths crossover on Hamming graphs (hereinafter named *all-Hamming-paths* crossover) is not a geometric crossover (Theorem 5.8) but it is a monotonic recombination P-structure (Theorems 5.9–5.10). First, Lemma 5.1 needs to ensure that all-Hamming-paths crossover outputs as offspring all vertices of a Hamming graph for different parents.

Lemma 5.1 ([49])**.** Let x and y be distinct vertices of any n-dimensional q-ary ($q \geq 2$) Hamming graph $H(n, q)$. Then, AllPaths(x, y) returns all vertices of the Hamming graph as offspring.

Proof. Let us show constructively that all vertices in the Hamming graph are visited by at least one path between x and y. List all vertices of $H(n, q)$ as a Grey code cyclic sequence $(v_1, v_2, \ldots, v_{q^n})$ such that any two consecutive vertices are at Hamming distance $d_{\mathrm{H}}(v_i, v_{i+1}) = 1$, and the same holds for the first and last vertices $d_{\mathrm{H}}(v_1, v_{q^n}) = 1$ hence cyclic. Since x and y are distinct Hamming sequences, there exist distinct v_i and v_j such that $(v_1, \ldots, v_i = x, \ldots, v_j = y, \ldots, v_{q^n})$. Then, $p_1 \stackrel{\text{def}}{=} (v_i = x, v_{i+1}, \ldots, v_{j-1}, v_j = y)$ and $p_2 \stackrel{\text{def}}{=} (v_j = y, v_{j+1}, \ldots, v_{q^n}, v_1, \ldots, v_{i-1}, v_i = x)$ are disjoint paths between x and y because: both p_1 and p_2 start or end at x and y (so do not form a cycle), each pair of consecutive vertices is an edge since they are at Hamming distance one, and the paths are independent because the only common vertices are the start and end vertices. Therefore, every vertex in $H(n, q)$ is visited by p_1 or p_2. Figure 5.3 shows an example of two such paths. □

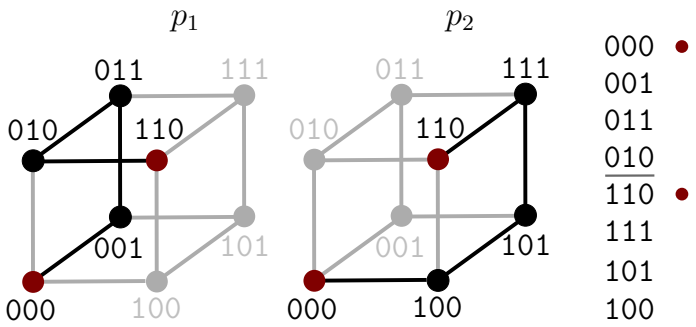


Figure 5.3. Two disjoint paths p_1 and p_2 on a three-dimensional binary Hamming graph $H(3, 2)$ by ordering its vertices as a Grey code cyclic sequence.

Theorem 5.8 (AllPaths$(H(n, q)) \in \overline{\mathcal{GX}}$ [49])**.** The all-Hamming-paths crossover, denoted AllPaths$(H(n, q))$, is not a geometric crossover.

Proof. Let us proof by counterexample that AllPaths$(H(n, q))$, AllPaths for short, does not fulfil the convergence inbreeding property (Proposition 3.3) in general. Let parents $x = \texttt{000}$ and $y = \texttt{001}$ for the three-dimensional binary Hamming graph $H(3, 2)$. Then, the second parent y is generated again by recombining the first parent x and the offspring $\texttt{111} \in$ AllPaths(x, y): $y \in$ AllPaths$(x, \texttt{111})$. Therefore, all-Hamming-paths is not a geometric crossover. □

Theorem 5.9 (AllPaths$(H(n, q)) \in \mathcal{RP}$ [49])**.** Let V be the vertex set of any n-dimensional q-ary ($q \geq 2$) Hamming graph $H(n, q)$. Then, $\big(V,$ AllPaths$(H(n, q))\big)$ is a recombination P-structure.

Proof. Let us prove that the all-Hamming-paths crossover fulfils the axioms of recombination P-structures.

(I) *Fix-point*: $\textsc{AllPaths}(x, x) = \{x\}$. It holds because the only path that can begin and end at offspring x is x itself. Otherwise cycles can be formed, and AllPaths only returns paths not cycles.

(II) *Symmetry*: $\textsc{AllPaths}(x, y) = \textsc{AllPaths}(y, x)$. It holds because AllPaths is an interval function [17], which by Definition 4.2 are symmetric.

(III) *Null-recombination*: $\{x, y\} \subseteq \textsc{AllPaths}(x, y)$. It holds because of the extensitivity axiom of interval functions (Definition 4.2), and AllPaths is an interval function [17].

(IV) *Size-monotonicity*: if z is offspring of $\textsc{AllPaths}(x, y)$, then $|\textsc{AllPaths}(x, z)| \leq |\textsc{AllPaths}(x, y)|$. If $x = y$, it is trivial because there is only one offspring: $z = x = y$. If $x \neq y$, there are two cases. Case 1: if $z = x$, then $|\textsc{AllPaths}(x, x)| = 1 \leq |\textsc{AllPaths}(x, y)|$. Case 2: if $z \neq x$, then AllPaths returns offspring in all possible paths between x and z, so all vertices are visited by at least one path (Lemma 5.1). Hence $|\textsc{AllPaths}(x, z)| = q^n \leq q^n = |\textsc{AllPaths}(x, y)|$, where q^n is the number of vertices of a Hamming graph. □

Theorem 5.10 ($\textsc{AllPaths}(H(n, q)) \in \mathcal{RP}$-monotonic)**.** Let V be the vertex set of any n-dimensional q-ary ($q \geq 2$) Hamming graph $H(n, q)$. Then, the recombination P-structure $\big(V, \textsc{AllPaths}(H(n, q))\big)$ is monotonic.

Proof. To be a monotonic recombination P-structure, AllPats needs to satisfy $\textsc{AllPaths}(u, v) \subseteq \textsc{AllPaths}(x, y)$, for all parents $x, y \in V$ and for all offspring $u, v \in \textsc{AllPaths}(x, y)$:

1. If $x = y$, then $\textsc{AllPaths}(x, y) = \textsc{AllPaths}(x, x) = \{x\}$. So $\textsc{AllPaths}(u, v) = \{x\} \subseteq \{x\} = \textsc{AllPaths}(x, x)$.

2. If $x \neq y$, then AllPaths returns all vertices of $H(n, q)$ by Lemma 5.1, that is $\textsc{AllPaths}(x, y) = V$. So $\textsc{AllPaths}(u, v) \subseteq \textsc{AllPaths}(x, y) = V$ since $\textsc{AllPaths}(u, v)$ is either a single vertex of V when u and v are identical, or the whole vertex set otherwise. □

Besides classification purposes, why consider the all-Hamming-paths crossover? From the previous Example 5.3 it is clear that all-Hamming-paths either outputs a single offspring solution that equals its parents or outputs the entire set of solutions

(i.e. all vertices of the Hamming graph). Hence it is impractical for most EAs. Nevertheless, all-Hamming-paths will be helpful in Part II to establish key behavioural differences between EAs using geometric crossovers and recombination P-structures.

5.4.4 Intersecting-balls

This section defines a generic crossover, not bound to a specific search space, called *intersecting-balls* (Definition 5.5) and shows that, for Hamming metric spaces, it is a non-geometric crossover (Theorem 5.11) and a non-monotonic recombination P-structure (Lemma 5.2 and Theorems 5.12–5.13).

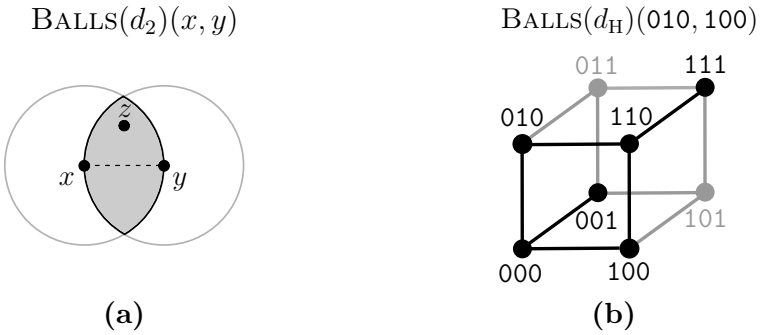


Figure 5.4. Intersecting-balls crossover in two-dimensional Euclidean metric space $(\mathbb{R}^2, d_2)$ in Figure 5.4a ('shaded region') and in three-dimensional binary Hamming space $(\mathcal{H}_2^3, d_{\mathrm{H}})$ in Figure 5.4b ('black lines').

Intuitively, for given parents in an arbitrary metric space, intersecting-balls outputs as offspring set all the points in the metric space whose distance from the parents does not exceed the distance between the parents themselves. In Euclidean spaces, the offspring set is seen by intersecting the two Euclidean metric closed balls centred at each parent and using their mutual distance as radius (Figure 5.4a), analogously for Hamming spaces (Figure 5.4b). Appendix A.3 provides a software implementation of intersecting-Hamming-balls.

Definition 5.5 (**Intersecting-balls crossover** [49])**.** Let (X, d) be any metric space and arbitrary parents $x, y \in X$. Then, the *intersecting-balls crossover* is defined as $\textsc{Balls}(d) : X \times X \to \mathcal{P}(X)$, $(x, y) \mapsto \bar{B}_d(x, d(x, y)) \cap \bar{B}_d(y, d(y, x))$ parametrised with metric d.

Definition 5.6 (**Intersecting-Hamming-balls crossover** [49])**.** Let $(\mathcal{H}_q^n, d_{\mathrm{H}})$ be any n-dimensional q-ary ($q \geq 2$) Hamming metric space. Then, the *Hamming intersecting-balls crossover* is $\textsc{Balls}(d_{\mathrm{H}})(x, y) \stackrel{\text{def}}{=} \bar{B}_{d_{\mathrm{H}}}(x, d_{\mathrm{H}}(x, y)) \cap \bar{B}_{d_{\mathrm{H}}}(y, d_{\mathrm{H}}(y, x))$ for any parents $x, y \in \mathcal{H}_q^n$.

Theorem 5.11 ($\textsc{Balls}(d_{\mathrm{H}}) \in \overline{\mathcal{GX}}$ [49])**.** The intersecting-Hamming-balls crossover, denoted $\textsc{Balls}(d_{\mathrm{H}})$, is not a geometric crossover.

Proof. Let us proof by counterexample that $\textsc{Balls}(d_{\mathrm{H}})$ does not fulfil in general the convergence inbreeding property (Proposition 3.3). Let parents `010` and `100` in three-dimensional binary Hamming metric space $(\mathcal{H}_2^3, d_{\mathrm{H}})$. Then, offspring `001` $\in \textsc{Balls}(d_{\mathrm{H}})(\texttt{010}, \texttt{100})$, but the second parent `100` is generated by recombining the first parent `010` and the offspring `001`: `100` $\in \textsc{Balls}(d_{\mathrm{H}})(\texttt{010}, \texttt{001})$. Therefore, $\textsc{Balls}(d_{\mathrm{H}})$ is not a geometric crossover. □

Intersecting-Hamming-balls crossover $\textsc{Balls}(d_{\mathrm{H}})$ is a recombination P-structure by Theorem 5.12. Compared with the fix-point, symmetry and null-recombination axioms, proving size-monotonicity is more complicated. Size-monotonicity states: if $z \in \textsc{Balls}(d)(x, y)$, then $|\textsc{Balls}(d)(x, z)| \leq |\textsc{Balls}(d)(x, y)|$. The main difficulty is $\textsc{Balls}(d)(x, z) \not\subseteq \textsc{Balls}(d)(x, y)$, in general; however, notice that does not disprove size-monotonicity necessarily: $A \not\subseteq B \implies |A| > |B|$ for arbitrary sets A, B. The overall idea to proof size-monotonicity is summarised by the following three observations that constitute the proof's main steps:

1. In principle, z may be anywhere in the intersection of the two Hamming balls given by $\textsc{Balls}(d_{\mathrm{H}})(x, y)$. But, regardless of z, there is always a $z' \in [x, y]_{d_{\mathrm{H}}}$ such that $d_{\mathrm{H}}(x, z) = d_{\mathrm{H}}(x, z')$.

2. Hamming graphs are distance-transitive [14]. So whenever $d_{\mathrm{H}}(x, z) = d_{\mathrm{H}}(x, z')$, there is an automorphism ϕ such that $\phi(x) = x$ and $\phi(z) = z'$ (Figure 5.5). Every graph automorphism is an isomorphism by definition (see [59, ch. 14] or [14, appx. A.3]), so ϕ preserves the adjacency structure of the Hamming graph.

3. Lemma 5.2 shows, for all $z' \in [x, y]_{d_{\mathrm{H}}}$, $\textsc{Balls}(d_{\mathrm{H}})(x, z') \subseteq \textsc{Balls}(d_{\mathrm{H}})(x, y)$ holds.

Together, the previous three observations essentially mean that size-monotonicity of $\textsc{Balls}(d_{\mathrm{H}})(x, z)$ can be proved through the size-monotonicity of $\textsc{Balls}(d_{\mathrm{H}})(x, z')$, by relabelling with ϕ the vertices of a reference Hamming graph associated with $\textsc{Balls}(d_{\mathrm{H}})$.

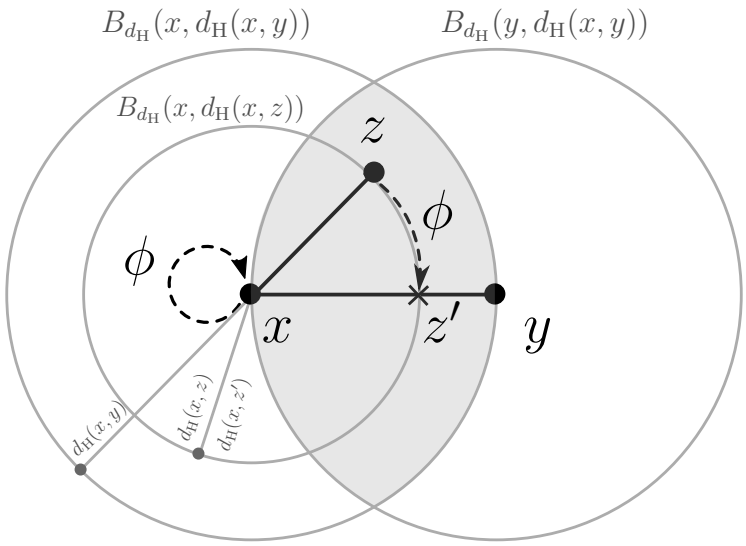


Figure 5.5. Diagram of the automorphism ϕ used in Theorem 5.12 to prove size-monotonicity of the intersecting-Hamming-balls crossover. Relying on distance-transitivity of Hamming graphs, there is such ϕ mapping an offspring $z \in \textsc{Balls}(d_H)(x,y)$ to another one $z' \in \textsc{Balls}(d_H)(x,y)$, at the same distance from x, where $z' \in [x,y]_{d_H}$. Size-monotonicity follows immediately as a consequence of $\textsc{Balls}(d_H)(x,z') \subseteq \textsc{Balls}(d_H)(x,y)$ using Lemma 5.2. This diagram is an auxiliary visual aid, not a true illustration of such automorphism in Hamming space.

Lemma 5.2 ([49])**.** Let (X,d) be any metric space. Then, for all $x,y \in X$ it holds: if $z \in [x,y]_d$, then $\textsc{Balls}(d)(x,z) \subseteq \textsc{Balls}(d)(x,y)$.

Proof. Note first that if $z \in [x,y]_d$, then naturally $z \in \textsc{Balls}(d)(x,y)$. We need to prove that if $s \in \textsc{Balls}(d)(x,z)$, then $s \in \textsc{Balls}(d)(x,y)$. By Definition 5.5 of intersecting-balls crossover:

$$z \in \textsc{Balls}(d)(x,y) \iff \Big(d(x,z) \leq d(x,y)\Big) \text{ and } \Big(d(y,z) \leq d(x,y)\Big) \ , \tag{5.1}$$

$$s \in \textsc{Balls}(d)(x,z) \iff \Big(d(x,s) \leq d(x,z)\Big) \text{ and } \Big(d(z,s) \leq d(x,z)\Big) \ . \tag{5.2}$$

Hence $d(x,s) \leq d(x,z) \leq d(x,y)$, and therefore $s \in \bar{B}_d(x, d(x,y))$. To complete the proof we need to show $s \in \bar{B}_d(y, d(y,x))$. That is, to show $d(y,s) \leq d(x,y)$ holds. From the assumption $z \in [x,y]_d$

$$d(x,y) = d(x,z) + d(z,y) \ , \tag{5.3}$$

and from the triangle inequality of metric spaces

$$d(y,s) \leq d(y,z) + d(z,s) \ . \tag{5.4}$$

Combining Equations 5.3–5.4

$$d(y,s) \leq d(y,z) + d(z,s) \leq d(y,z) + d(x,z) = d(x,y) \tag{5.5}$$

because $d(z,s) \leq d(x,z)$ from Equation 5.2, and because of symmetry $d(y,z) = d(z,y)$ in metric spaces. Hence from Equation 5.5 it holds $s \in \bar{B}_d(y, d(y,x))$, so $s \in \textsc{Balls}(d)(x,y)$. Therefore, if $s \in \textsc{Balls}(d)(x,z)$, then $s \in \textsc{Balls}(d)(x,y)$. □

Theorem 5.12 ($\textsc{Balls}(d_{\mathrm{H}}) \in \mathcal{RP}$ [49])**.** Let $(\mathcal{H}_q^n, d_{\mathrm{H}})$ be any reference n-dimensional q-ary ($q \geq 2$) Hamming metric space. Then, $(\mathcal{H}_q^n, \textsc{Balls}(d_{\mathrm{H}}))$ is a recombination P-structure.

Proof. Let us prove that the intersecting-Hamming-balls crossover fulfils the axioms of recombination P-structures.

(I) *Fix-point*: $\textsc{Balls}(d_{\mathrm{H}})(x,x) = \{x\}$. In metric spaces $d_{\mathrm{H}}(x,y) = 0$ iff $x = y$. Therefore, $\textsc{Balls}(d_{\mathrm{H}})(x,x) = \bar{B}_{d_{\mathrm{H}}}(x, d_{\mathrm{H}}(x,x)) \cap \bar{B}_{d_{\mathrm{H}}}(x, d_{\mathrm{H}}(x,x)) = \bar{B}_{d_{\mathrm{H}}}(x,0) \cap \bar{B}_{d_{\mathrm{H}}}(x,0) = \{x\} \cap \{x\} = \{x\}$.

(II) *Symmetry*: $\textsc{Balls}(d_{\mathrm{H}})(x,y) = \textsc{Balls}(d_{\mathrm{H}})(y,x)$. Follows from Definition 5.6 and commutativity of intersection: $A \cap B = B \cap A$.

(III) *Null-recombination*: $\{x,y\} \subseteq \textsc{Balls}(d_{\mathrm{H}})(x,y)$. Since $d_{\mathrm{H}}(x,x) = 0 \leq d_{\mathrm{H}}(x,y)$, we know $x \in \bar{B}_{d_{\mathrm{H}}}(x, d_{\mathrm{H}}(x,y))$. Since $d_{\mathrm{H}}(x,y) \leq d_{\mathrm{H}}(x,y)$, we know as well $y \in \bar{B}_{d_{\mathrm{H}}}(x, d_{\mathrm{H}}(x,y))$. Therefore, $\{x,y\} \subseteq \bar{B}_{d_{\mathrm{H}}}(x, d_{\mathrm{H}}(x,y))$. By symmetry, $\{x,y\} \subseteq \bar{B}_{d_{\mathrm{H}}}(y, d_{\mathrm{H}}(y,x))$. So $\{x,y\} \subseteq \left(\bar{B}_{d_{\mathrm{H}}}(x, d_{\mathrm{H}}(x,y)) \cap \bar{B}_{d_{\mathrm{H}}}(y, d_{\mathrm{H}}(y,x))\right)$ $= \textsc{Balls}(d_{\mathrm{H}})(x,y)$.

(IV) *Size-monotonicity*: if z is offspring of $\textsc{Balls}(d_{\mathrm{H}})(x,y)$, then $|\textsc{Balls}(d_{\mathrm{H}})(x,z)|$ $\leq |\textsc{Balls}(d_{\mathrm{H}})(x,y)|$. The proof follows from: (a) $\forall z \in \textsc{Balls}(d_{\mathrm{H}})(x,y)$ there is a $z' \in [x,y]_{d_{\mathrm{H}}}$ where $d_{\mathrm{H}}(x,z) = d_{\mathrm{H}}(x,z')$, so that using (b) and (c) below, we have $|\textsc{Balls}(d_{\mathrm{H}})(x,z)| = |\textsc{Balls}(d_{\mathrm{H}})(x,z')| \leq |\textsc{Balls}(d_{\mathrm{H}})(x,y)|$.

(a) We prove that: $\forall x,y \in \mathcal{H}_q^n$, if $z \in \textsc{Balls}(d_{\mathrm{H}})(x,y)$, then $\exists z' \in [x,y]_{d_{\mathrm{H}}}$ such that $d_{\mathrm{H}}(x,z) = d_{\mathrm{H}}(x,z')$. Assuming $z \in \textsc{Balls}(d_{\mathrm{H}})(x,y)$, we know $d_{\mathrm{H}}(x,z) \leq d_{\mathrm{H}}(x,y)$. Because the Hamming graph is connected and undirected, there exists at least one path between x and y, in particular a shortest path $[x,y]_{d_{\mathrm{H}}}$. Because it is unweighted, each integer distance unit between x and y has a corresponding $z' \in [x,y]_{d_{\mathrm{H}}}$. Since $d_{\mathrm{H}}(x,z) \leq d_{\mathrm{H}}(x,y)$ and $d_{\mathrm{H}}(x,z') \leq d_{\mathrm{H}}(x,y)$, it is always possible to find a $z' \in [x,y]_{d_{\mathrm{H}}}$ such that $d_{\mathrm{H}}(x,z) = d_{\mathrm{H}}(x,z')$.

(b) From Lemma 5.2 follows that if $z' \in [x,y]_{d_H}$, then $|\textsc{Balls}(d_H)(x,z')| \leq |\textsc{Balls}(d_H)(x,y)|$.

(c) We prove that: for all $x,z,z' \in \mathcal{H}_q^n$, if $d_H(x,z) = d_H(x,z')$, then $|\textsc{Balls}(d_H)(x,z)| = |\textsc{Balls}(d_H)(x,z')|$. It follows from the distance-transitivity of Hamming graphs (Theorem 9.2.1 in [14]). That means if $d_H(x,z) = d_H(x,z')$, then there is an automorphism $\phi : \mathcal{H}_q^n \to \mathcal{H}_q^n$ where $\phi(x) = x$ and $\phi(z) = z'$ (see Figure 5.5). Graphs' automorphisms are isomorphisms by definition, preserving the graph adjacency structure (see [59, ch. 14] or [14, appx. A.3]). Because ϕ is an adjacency-preserving map, ϕ cannot alter the number of common neighbours of x and z at distance $d_H(x,z)$, namely $|\bar{B}_{d_H}(x, d_H(x,z)) \cap \bar{B}_{d_H}(z, d_H(z,x))|$. Hence $|\textsc{Balls}(d_H)(x,z)| = \left|\textsc{Balls}(d_H)\Big(\phi(x),\phi(z)\Big)\right| = |\textsc{Balls}(d_H)(x,z')|$. □

Theorem 5.13 ($\textsc{Balls}(d_H) \in$ co-$\mathcal{RP}$-monotonic)**.** Let $(\mathcal{H}_q^n, d_H)$ be any n-dimensional q-ary ($q \geq 2$) Hamming metric space. Then, the recombination P-structure $(\mathcal{H}_q^n, \textsc{Balls}(d_H))$ is not monotonic.

Proof. To be a monotonic recombination P-structure, $\textsc{Balls}(d_H)$ needs to satisfy $\textsc{Balls}(d_H)(u,v) \subseteq \textsc{Balls}(d_H)(x,y)$, $\forall x,y \in \mathcal{H}_q^n$ and $\forall u,v \in \textsc{Balls}(d_H)(x,y)$. The next counterexample proves it is false in general. Let parents $x = \texttt{010}$ and $y = \texttt{100}$ as in Figure 5.2b. Let offspring $u = \texttt{100}$, $v = \texttt{111}$ in $\textsc{Balls}(d_H)(x,y)$. Then, $\textsc{Balls}(d_H)(u,v) \not\subseteq \textsc{Balls}(d_H)(x,y)$ since $\texttt{101} \in \textsc{Balls}(d_H)(u,v)$ but $\texttt{101} \notin \textsc{Balls}(d_H)(x,y)$. □

5.4.5 Identity

The *identity* crossover Id (Definition 4.4), also called *discrete metric geometric crossover* [100], accepts any pair of parents x and y as input but just outputs those parents as offspring. Therefore, it is a fix-point operator under repeated applications: $\{x,y\} = \textsc{Id}(x,y) = \textsc{Id}(\textsc{Id}(x,y)) \ldots$. Although Id is impractical for EAs since it cannot produce new solutions, it is relevant for the crossover classification because:

- Every crossover ξ whose offspring sets do not include the identity, that is $\textsc{Id}(x,y) = \{x,y\} \not\subseteq \xi(x,y)$, cannot be a recombination P-structure since ξ immediately violates the null-recombination axiom requiring $\{x,y\} \subseteq \xi(x,y)$.

- Even if a crossover's definition differs from Id, their structure may be equivalent in certain cases. For example, the structure of traditional crossovers

on strings (i.e. Hamming sequences), like uniform or one-point crossover, is equivalent to ID [139] when:

- The Hamming distance between parents is one. For instance, $d_{\mathrm{H}}(\mathtt{000}, \mathtt{001}) = 1$ and $\textsc{Uniform}(\mathtt{000}, \mathtt{001}) = \{\mathtt{000}, \mathtt{001}\} = \textsc{Id}(\mathtt{000}, \mathtt{001})$.
- The crossover is an involution, namely a function that is its own inverse $f(f(x)) = x$. For instance, consider the one-point crossover function $\textsc{OnePoint}(i)$ at crossover point $i = 1$. Then, $\textsc{OnePoint}(1)(\mathtt{000}, \mathtt{111}) = \{\mathtt{011}, \mathtt{100}\}$, but $\textsc{OnePoint}(1)(\mathtt{011}, \mathtt{100}) = \{\mathtt{000}, \mathtt{111}\}$, hence:

$$\textsc{OnePoint}(1)\Big(\overbrace{\textsc{OnePoint}(1)(\mathtt{000}, \mathtt{111})}^{\mathtt{011},\,\mathtt{100}}\Big) = \{\mathtt{000}, \mathtt{111}\} = \textsc{Id}(\mathtt{000}, \mathtt{111}) \ .$$

If two crossovers belong to the same class, their structure is equivalent to the extent that both share the axioms of such class. Therefore, knowing which (sub)class ID belongs to can help distinguish crossovers with a similar structure, like ONEPOINT and UNIFORM, from those dissimilar. Theorem 5.14 proves ID to be a finite-incomplete geometric crossover that is simultaneously a monotonic and geometric recombination P-structure.

Theorem 5.14 ($\textsc{Id} \in$ co-$\mathcal{GX}$-complete$_{fin}$, $\mathcal{RP}$-geometric, $\mathcal{RP}$-monotonic)**.** Let (X, d) be any finite graphic metric space and any parents $x, y \in X$. Then, $\textsc{Id}(x, y)$ is a finite-incomplete geometric crossover, and $(X, \textsc{Id})$ is both a geometric and monotonic recombination P-structure.

Proof. The proof is trivial because $\textsc{Id}(x, y) = \{x, y\}$ for all $x, y \in X$:

1. ID is a finite-incomplete geometric crossover (Definition 3.3) since it is a subset of the metric segment between parents: $\textsc{Id}(x, y) = \{x, y\} \subseteq [x, y]_d$.
2. $(X, \textsc{Id})$ fulfils all axioms of recombination P-structures (Definition 4.3): fixpoint, $\textsc{Id}(x, x) = \{x\}$, symmetry, $\textsc{Id}(x, y) = \textsc{Id}(y, x) = \{x, y\}$, null-recombination, $\{x, y\} \subseteq \textsc{Id}(x, y) = \{x, y\}$, and size-monotonicity $|\textsc{Id}(x, z)| \leq |\textsc{Id}(x, y)|$ since $\textsc{Id}(x, z) \subseteq \textsc{Id}(x, y)$ for all $z \in \textsc{Id}(x, y)$.
3. $(X, \textsc{Id})$ is a geometric recombination P-structure (Definition 4.8) since metric segments are shortest-paths on graphic metric spaces and always include their extremes: $\textsc{Id}(x, y) = \{x, y\} \subseteq [x, y]_d$.
4. $(X, \textsc{Id})$ is a monotonic recombination P-structure (Definition 4.9) since for all $u, v \in \textsc{Id}(x, y) = \{x, y\}$: $\textsc{Id}(u, v) \subseteq \textsc{Id}(x, y)$. □

5.4.6 Geometric Crossovers vs Recombination P-structures

The relationships between the geometric crossovers and recombination P-structures (sub)classes (Table 5.2) must be determined to prove the classification of crossovers overviewed in Section 5.2. Relying on Section 5.3 and Sections 5.4.1–5.4.5, Theorem 5.15 proves a non-exhaustive list of the main relationships between such classes that form the classification. Later, Section 5.5 uses it to examine two concrete aspects of geometric crossovers: high-locality [144] and inbreeding properties [103].

Theorem 5.15 (**Class relationships** [49])**.** Given the universal class $\mathcal{U}$ of all crossovers, consider the (sub-)classes of: geometric crossovers $\mathcal{GX}$, finite-complete geometric crossovers $\mathcal{GX}$-complete$_{fin}$, recombination P-structures $\mathcal{RP}$, geometric recombination P-structures $\mathcal{RP}$-geometric and monotonic recombination P-structures $\mathcal{RP}$-monotonic. The following relationships hold:

(a) $\overline{\mathcal{GX} \cup \mathcal{RP}} \neq \emptyset$. *The class of crossovers that are neither geometric nor recombination P-structures is not empty.*

(b) $\mathcal{GX} \neq \mathcal{RP}$. *Geometric crossovers and recombination P-structures are distinct classes.*

(c) $\mathcal{RP} \cap \overline{\mathcal{GX}} \neq \emptyset$. *The class of crossovers that are recombination P-structures but not geometric crossovers is not empty.*

(d) $\mathcal{GX}$-complete$_{fin} \subset \mathcal{RP}$-geometric. *Finite-complete geometric crossovers are a strict subclass of geometric recombination P-structures.*

(e) $\mathcal{GX}$-complete$_{fin} \not\subseteq \mathcal{RP}$-monotonic. *Finite-complete geometric crossovers are not a subclass of monotonic recombination P-structures.*

(f) $\mathcal{RP}$-geometric $= \mathcal{GX} \cap \mathcal{RP}$. *Geometric recombination P-structures are a subclass of crossovers that are both geometric crossovers and recombination P-structures.*

(g) $\mathcal{RP}$-monotonic $\not\subseteq \mathcal{GX} \cap \mathcal{RP}$. *Monotonic recombination P-structures are not a subclass of crossovers that are both geometric crossovers and recombination P-structures.*

(h) $\mathcal{RP}$-geometric $\neq \mathcal{RP}$-monotonic. *Geometric recombination P-structures and monotonic recombination P-structures are distinct subclasses.*

(i) $\mathcal{RP}$-geometric $\subset \mathcal{RP}$. *Geometric recombination P-structures are a strict subclass of recombination P-structures.*

(j) $\mathcal{RP}$-monotonic $\subset \mathcal{RP}$. *Monotonic recombination P-structures is a strict subclass of recombination P-structures.*

Proof. (a) Koza's subtree swap and (symmetric) Davis's order crossovers are non-geometric (see Remark 5.1 and [103]). Theorems 5.6–5.7 prove they are neither recombination P-structures. Therefore, Koza, Davis and sym-Davis belong to $\overline{\mathcal{GX} \cup \mathcal{RP}}$.

(b) Asymmetric one-point crossover is a geometric crossover but not a recombination P-structure due to its asymmetry (Theorem 5.5).

(c) Theorems 5.11–5.12 prove that intersecting-Hamming-balls crossover is not a geometric crossover but is a recombination P-structure. Theorems 5.8–5.9 prove it analogously for all-Hamming-paths crossover.

(d) Theorem 5.1 proves that $\mathcal{GX}\text{-complete}_{fin} \subseteq \mathcal{RP}$. Since finite-complete geometric crossovers produce offspring in shortest-paths they are also geometric recombination P-structures by Definition 4.8. Hence $\mathcal{GX}\text{-complete}_{fin} \subseteq \mathcal{RP}$-geometric. The strict inclusion $\mathcal{GX}\text{-complete}_{fin} \subset \mathcal{RP}$-geometric follows because the one-point (OnePoint) and identity (Id) crossovers are geometric recombination P-structures but not finite-complete geometric crossovers (Theorems 5.4 and 5.14).

(e) In general, not all metric segments fulfil the monotonicity property of monotonic recombination P-structures (Definition 4.9). Figure 5.6 presents a counterexample. Hence not all finite-complete geometric crossovers are monotonic recombination P-structures.

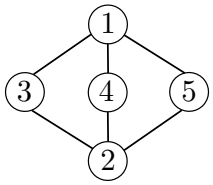


Figure 5.6. A $K_{2,3}$ bipartite graph where every finite-complete geometric crossover (i.e. metric segment) does not fulfil Definition 4.9 of monotonic recombination P-structures. That is, there exist vertices x, y and $u, v \in [x, y]_d$ such that $[u, v]_d \not\subseteq [x, y]_d$, d being the metric induced by $K_{2,3}$. For instance, $[1, 2]_d = \{1, 3, \underline{4}, 5, 2\} \not\subseteq \{3, 1, 2, 5\} = [3, 5]_d$.

(f) This is trivially Definition 4.8: geometric recombination P-structures are recombination P-structures that produce offspring in shortest paths between parents, thus being geometric crossovers (Definition 3.3).

(g) Although monotonic recombination P-structures are a subclass of recombination P-structures, not all of them are geometric crossovers: all-Hamming-paths crossover is one example (Theorems 5.8 and 5.10).

(h) All-Hamming-paths crossover is a monotonic (Theorem 5.10) but not geometric recombination P-structure since all-Hamming-paths is not a geometric crossover (Theorem 5.8). Conversely, one-point crossover is a geometric recombination P-structure but not monotonic (Theorem 5.4).

(i) By Definition 4.8, geometric recombination P-structures are a subclass of recombination P-structures. The strict inclusion $\mathcal{RP}$-geometric $\subset \mathcal{RP}$ follows because not all recombination P-structures are geometric. For instance, the intersecting-Hamming-balls crossover is in $\mathcal{RP}$ (Theorem 5.12) but not in $\mathcal{RP}$-geometric because it is not a geometric crossover (Theorem 5.11).

(j) By Definition 4.9, monotonic recombination P-structures are a subclass of recombination P-structures. The strict inclusion $\mathcal{RP}$-monotonic $\subset \mathcal{RP}$ holds because not all recombination P-structures are monotonic, for example one-point and intersecting-Hamming-balls crossovers (Theorems 5.4 and 5.13). □

The reasons that justify the significance of Theorem 5.15, and the crossover classification overall, concern: crossover structure, crossover design, and EAs' behaviour and performance.

Crossover Structure

Individually, the GF and ELT provide a partial view on the structure of crossovers, the former focusing only on geometric crossovers and the latter only on recombination P-structures. The crossover classification broadens these views by incorporating geometric crossovers and recombination P-structures all together. As a result, the GF and ELT are extended as follows:

- The coarse classification of crossovers into geometric and non-geometric is enhanced by the class and subclasses of recombination P-structures, which not necessarily coincide with geometric crossovers, enabling thus a more fine-grained classification of crossovers.

- Geometric crossovers are confined to metric spaces, whereas recombination P-structures may exist in non-metric spaces since they do not require metrics. So the crossover classification facilitates comparing crossovers between metric

and non-metric spaces through the axioms that define the (sub)classes. Furthermore, recombination P-structures are a bridge to other abstractions such as interval or transit functions and betweenness relations [18, 107, 146], which have been well studied and are potentially useful to gain further insight into geometric crossovers' structure.

- For the first time, ELT has a proper classification of crossovers. This classification reveals an existing but unnoticed overlap between geometric crossovers and various classes of recombination P-structures, bringing to light two important aspects left unclear in ELT. First, recombination P-structures are not a purely theoretical crossover class because they include potentially many practical geometric crossovers. Secondly, the search space structure of geometric recombination P-structures can be understood, like geometric crossovers, in intuitive geometrical terms via finite metric spaces as a simpler alternative to hypergraphs.

Crossover Design

Problem-specific representations and search operators are most profitable to achieve optimal algorithm performance [4, 96]. Yet when the problem requires unfamiliar representations, it is generally difficult to decide which search operators, crossovers in particular, should be used. This motivated the GF to propose a formal representation-independent strategy of designing crossovers called principled design (Section 3.3). By contrast, although ELT may provide valuable insight on design of crossovers (and EAs) [139, 140], it is unclear how ELT guides practitioners to select a specific crossover operator based on the general Definition 4.3 of recombination P-structures. That is, ELT does not provide a method for crossover design.

The crossover classification makes clear that if a crossover belongs to $\mathcal{GX} \cap \mathcal{RP}$, then it can be treated both as a geometric crossover ($\mathcal{GX}$) and a recombination P-structure ($\mathcal{RP}$). Hence the crossover classification reveals a fundamental aspect of crossover design in ELT: the recombination P-structure class contains crossovers that can be designed according to the GF's principled design. Or, put differently, the GF's principled design is compatible with certain recombination P-structures. Therefore, principled crossover design is possible in ELT. Furthermore, the limits of principled design are clarified as follows. Recall that principled design entails finding a suitable distance or metric to derive specific crossovers (Section 3.3). Clearly, such design strategy is inapplicable to crossovers whose structure cannot be formalised using a notion of distance, namely in non-metric spaces. So principled design is of

little or no use for crossovers in $\mathcal{RP} \cap \overline{\mathcal{GX}}$, namely recombination P-structures that are non-geometric crossovers (e.g. intersecting-Hamming-balls and all-paths), since recombination P-structures do not require distances in general.

Behaviour and Performance of EAs

The crossover classification by itself does not state anything about EA performance since it does not consider problem-algorithm class pairs, thus NFL prevails [155]. For the same reason, it does neither tell us how practical a crossover is on some problem. For example, Koza's subtree swap crossover is useful in some situations[2] [4, 86], yet it falls outside the classes of geometric crossovers and recombination P-structures, rendering the GF and ELT of little use to further study Koza's crossover. How then is the crossover classification relevant here?

The crossover classification does state the structure of geometric crossovers and recombination P-structures. This is relevant since the GF holds that the essential behavioural differences between EAs lie in the solutions' representation and search operators, making search space structures a natural choice to classify EAs. For geometric crossovers, their structure is reflected in EAs without mutation as an abstract form of convex search behaviour [100, 101]. Surprisingly, such form of convex search is extensible to ELT for the recombination P-structure class (as Chapter 8 will show), and the reason it is possible is precisely that recombination P-structures are a superclass of certain geometric crossovers as the classification shows. Therefore, the classification underpins not just the structure of crossovers but also their effect in EAs, stressing thus its significance as a foundation to unify the GF and ELT.

5.5 Exploiting the Crossover Classification

This section focuses on two properties related to geometric crossovers as a case study, relying on the classification of crossovers in Section 5.4. First, Section 5.5.1 shows why geometric crossovers and highly local crossovers are not equivalent in general [144] by checking the existence of a non-geometric yet highly local crossover in the classification. Afterwards, Section 5.5.2 tackles an open problem on the inbreeding properties (Section 3.5) by finding a new subclass of recombination P-structures that fulfils them.

[2]For instance, Koza's subtree swap may help prevent premature convergence of populations in EAs [86] since, in contrast with traditional genetic crossovers on strings, it can produce offspring different from parents even when these are identical.

5.5.1 High-locality

In contrast with *locality of representations* [123], which concerns the design of effective genotype-phenotype maps, *locality of search operators* concerns the design of effective search operators [124, 144]. In particular, locality of crossovers is a property describing the similarity between parents and offspring in terms of distances (Definition 5.7). It affects the performance of EAs: designing crossovers with high locality is desirable because dissimilarity between parents and offspring, present in crossovers with low locality, potentially leads an EA to random search [124, 144]. High-locality was also proposed in [38] though under the name 'Guideline R 1' as a 'good' crossover design rule.

Definition 5.7 (**High locality** [124, 144])**.** Let (X, d) be any metric space. Then, a crossover ξ has *high locality* on (X, d), if for any parents $x, y \in X$ and any offspring $z \in \xi(x, y)$: $\max\{d(x, z), d(z, y)\} \leq d(x, y)$. That is, if the distances between parents and offspring never exceed the distance between parents themselves.

Geometric crossovers (Definition 3.3) and high locality (Definition 5.7) are considered equivalent in [124, 144], that is geometric crossovers have high locality; however, such equivalence is not formally proved. A priori it is unknown if there exists a crossover or a class in the crossover classification (Figure 5.1) with high locality. Accordingly, following the bottom-up strategy outlined in Section 5.2.1, Lemma 5.3 proves next that every intersecting-ball $\textsc{Balls}(d)$ crossover (Definition 5.5) has high locality, in particular the intersecting-Hamming-balls crossover (Definition 5.6).

Lemma 5.3 ([49])**.** Let (X, d) be any metric space and arbitrary parents $x, y \in X$. Consider the set of points with high locality $R(x, y) \stackrel{\text{def}}{=} \{z \mid \max\{d(x, z), d(z, y)\} \leq d(x, y)\}$. Then, $R(x, y)$ is identical to $\textsc{Balls}(d)(x, y)$.

Proof. Follows directly: $z \in R(x, y) \iff d(x, z) \leq d(x, y)$ and $d(z, y) \leq d(x, y)$ $\iff z \in \left(\bar{B}_d(x, d(x, y)) \cap \bar{B}_d(y, d(y, x))\right) \iff z \in \textsc{Balls}(d)(x, y)$. □

As a result of Lemma 5.3, and because intersecting-Hamming-balls is not geometric, Corollaries 5.1–5.2 show that geometric crossovers always have high locality but not all crossovers with high locality are geometric crossovers necessarily.

Corollary 5.1 (**High locality $\not\equiv$ geometricity**)**.** There exists a crossover with high locality that is not a geometric crossover.

Proof. The intersecting-Hamming-balls crossover has high locality (Lemma 5.3) but it is not a geometric crossover in Hamming metric spaces (Theorem 5.11). □

Corollary 5.2 (Geometricity $\not\Leftrightarrow$ High locality). Geometric crossovers have high locality for any metric space, but not all crossovers with high locality are geometric.

Proof. Due to Lemma 5.3, all intersecting-balls $\textsc{Balls}(d)$ crossovers (Definition 5.5) have high locality (Definition 5.7) regardless of the metric space. Now, notice for any metric space (X, d) and any $x, y \in X$, $[x, y]_d \subseteq \textsc{Balls}(d)(x, y)$ holds because: for all $z \in [x, y]_d$ we have $d(x, z) \leq d(x, y)$ and $d(z, y) \leq d(x, y)$, which means $z \in \textsc{Balls}(d)(x, y)$. Therefore, any geometric crossover ξ defined on any metric space (X, d) has high locality since $\xi(x, y) \subseteq [x, y]_d \subseteq \textsc{Balls}(d)(x, y)$. However, not all highly local crossovers are geometric due to Corollary 5.1. □

Corollaries 5.1–5.2 are significant for three reasons. First, they clarify that high locality and geometricity of crossovers are not equivalent concepts in general as stated in [124, 144] or assumed implicitly in [38]; rather, geometricity is a particular case of high locality. This matters because it shows that geometricity like high locality is a property favouring 'good' EA performance even if they do not coincide always. Secondly, it clarifies the following important aspect about the formal design of highly local crossovers. The classification in Section 5.4.4 shows that intersecting-Hamming-balls crossover is non-geometric; therefore GF's crossover principled design of geometric crossovers (Section 3.3) does not apply to highly local crossovers always. This limits the extent to which highly local crossovers can be designed across representations in a principled manner. Finally, the positive side is that out of all crossovers with high locality, geometric crossovers ξ happen to be a subset due to $\xi(x, y) \subseteq [x, y]_d \subseteq \textsc{Balls}(d)(x, y)$. High locality and geometricity coincide precisely when $[x, y]_d = \textsc{Balls}(d)(x, y)$ for a given metric d and all x and y. So, after all, there are certain highly local crossovers that can be designed in a principled manner: geometric crossovers!

5.5.2 Inbreeding Properties

The inbreeding properties can tell if a crossover is non-geometric without having to test a possibly infinite number of metrics which Definition 3.7 requires. Proving that a crossover is geometric is relatively easy for some search spaces by suggesting a metric for them and then verifying Definition 3.3. But it can be difficult in general, even in well known search spaces as Moraglio [100] points out for trees in GP. This brings us to the following question left open in the GF (Section 3.5).

Question 5.3. *Are the inbreeding properties sufficient conditions to guarantee that a crossover is geometric?*

If the answer to Question 5.3 was affirmative, then the inbreeding properties could be used to test geometricity of crossovers also. Why would it be useful? Besides knowing whether a crossover is geometric, one may want to know under which metric. But testing the inbreeding properties (if sufficient) would only tell us the former not the latter, for they are independent of metrics. However, testing the inbreeding properties is useful for classification purposes: to classify a crossover as geometric or non-geometric, it is only necessary to know if it is geometric or not! If a given crossover was geometric, and we did not know the distance, the inbreeding properties (if sufficient) would classify it as geometric without knowing the distance.

This section first makes some remarks about proving the insufficiency of the inbreeding properties to clarify why answering Question 5.3 is not as simple as one may expect. Then, Question 5.3 is formulated more precisely and partially answered adopting the top-down approach outlined in Section 5.2.1, explaining why it is only a partial answer.

Proving Insufficiency of the Inbreeding Properties: Remarks

To answer Question 5.3, one looks for a crossover (or class of crossovers) that fulfils the inbreeding properties but it is not a geometric crossover, which would prove the inbreeding properties insufficient. A simple approach could be to claim that: recombination P-structures are such class, by proving they fulfil all the inbreeding properties, and proving that only the subclass of geometric recombination P-structures is equivalent to geometric crossovers. There are two issues with this claim:

1. It is false that all recombination P-structures fulfil the inbreeding properties, for example all-Hamming-paths and intersecting-Hamming-balls crossovers fail the convergence inbreeding property (Sections 5.4.3–5.4.4).

2. It is false that the geometric recombination P-structures subclass is equivalent to the geometric crossovers class: the asymmetric one-point crossover is a geometric crossover but not a recombination P-structure (Theorem 5.5). Also, geometric recombination P-structures and finite-complete geometric crossovers are not equivalent neither: one-point and identity crossovers are geometric recombination P-structures but not finite-complete geometric crossovers (Sections 5.4.1 and 5.4.5).

Fortunately, there actually exist a subclass of recombination P-structures that fulfils all the inbreeding properties as shown next.

Proving Insufficiency of the Inbreeding Properties: A Partial Answer

Although not all recombination P-structures fulfil the inbreeding properties, it may be possible that a subclass of recombination P-structures fulfils them and it contains non-geometric crossovers; thus proving the insufficiency of the inbreeding properties. This brings us to Question 5.4, which restates Question 5.3 in terms of insufficiency rather than sufficiency and more precisely.

Question 5.4. *Is there any subclass of recombination P-structures other than geometric recombination P-structures that satisfies all the inbreeding properties of geometric crossovers? If so, what is an example of such a recombination P-structure that is not a geometric crossover but fulfils the inbreeding properties?*

To answer the first part of Question 5.4, the top-down approach (Section 5.2.1) can be adopted and suggest a subclass of recombination P-structures with the desired inbreeding properties. Accordingly, Definition 5.8 introduces *strict size-monotonic recombination P-structures* as a new subclass of recombination P-structures by restricting the original size-monotonicity axiom. An example of a strict size-monotonic recombination P-structure is shown later in Example 5.4. Next, Lemma 5.4 proves that they are a subclass of recombination P-structures. Theorem 5.16 proves that they fulfil the inbreeding properties adapting the proof of Theorem 2 in [48].

Definition 5.8 (**Strict size-monotonic recomb. P-structure**)**.** Let X be any non-empty finite set and $\mathcal{R} : X \times X \to \mathcal{P}(X)$ mapping pairs of parents into offspring sets. Then, $(X, \mathcal{R})$ is a *strict size-monotonic recombination P-structure*, if $\forall x, y, z \in X$: (I) fix-point: $\mathcal{R}(x, x) = \{x\}$; (II) symmetry: $\mathcal{R}(x, y) = \mathcal{R}(y, x)$; (III) null-recombination: $\{x, y\} \subseteq \mathcal{R}(x, y)$; and (IV) strict size-monotonicity: if $z \in \mathcal{R}(x, y)$ and $z \neq y$, then $|\mathcal{R}(x, z)| < |\mathcal{R}(x, y)|$. The *class* of all strict size-monotonic recombination P-structures is denoted by $\mathcal{RP}$-strictsize.

Lemma 5.4. Let $(X, \mathcal{R})$ be a strict size-monotonic recombination P-structure. Then, $(X, \mathcal{R})$ is also a recombination P-structure.

Proof. The proof is trivial by Definition 5.8 since strict size-monotonicity is a particular case of size-monotonicity (Definition 4.3). □

Theorem 5.16 ($\mathcal{RP}$-strictsize **fulfils inbreeding properties**)**.** Let $(X, \mathcal{R})$ be any strict size-monotonic recombination P-structure. Then, $\mathcal{R}$ satisfies all the inbreeding properties of geometric crossovers: purity, convergence and partition.

Proof. Purity: Recombining one parent with itself can only produce the parent itself. Follows immediately from the fix-point axiom: $\forall x \in X$, $\mathcal{R}(x, x) = \{x\}$.

Convergence: Recombining one parent with one offspring cannot produce the other parent of that offspring, unless the offspring and the second parent coincide. Let offspring $z \in \mathcal{R}(x, y)$ for any parents $x, y \in X$, and let also $s \in \mathcal{R}(x, z)$. We want to prove that if $z \neq y$, then $s \neq y$. Assume $z \neq y$. Because $\mathcal{R}$ is a strict size-monotonic recombination P-structure,

$$|\mathcal{R}(x, z)| < |\mathcal{R}(x, y)| \quad . \tag{5.6}$$

Besides, either $s = z$ or $s \neq z$. If $s = z$, then immediately $s \neq y$ since $s = z \neq y$ and we are done. If $s \neq z$, then

$$|\mathcal{R}(x, s)| < |\mathcal{R}(x, z)| \tag{5.7}$$

because $\mathcal{R}$ is a strict size-monotonic recombination P-structure. Now, if s was allowed to be y, then $|\mathcal{R}(x, s)| = |\mathcal{R}(x, y)|$; however, by Equation 5.7

$$|\mathcal{R}(x, s)| = |\mathcal{R}(x, y)| < |\mathcal{R}(x, z)| \quad , \tag{5.8}$$

thus contradicting Equation 5.6. Therefore, $s \neq y$ for the $s \neq z$ case as well. Consequently, the only possibility left for $s = y$ is that $z = y$.

Partition: If z is a child of x and y, then the recombination of x and z, and the recombination of y and z, cannot produce a common grandchild s other than z. Let a child $z \in \mathcal{R}(x, y)$ for any parents $x, y \in X$. Let also grandchildren $s_1 \in \mathcal{R}(x, z)$ and $s_2 \in \mathcal{R}(z, y)$. We want to prove that if $s_1 \neq z$ or $s_2 \neq z$ (or both), that is if at least one grandchild is different from z, then $s_1 \neq s_2$. Let us clear all the trivial cases first.

- Trivial case 1: $z = y$. Then, $\mathcal{R}(x, z) = \mathcal{R}(x, y)$ and $\mathcal{R}(z, y) = \mathcal{R}(y, y)$, so $s_1 \in \mathcal{R}(x, y)$ and $s_2 \in \mathcal{R}(y, y) = \{y\}$. Hence $s_2 = y = z$, but notice $s_1 \neq z$ since at least one grandchild must be different from z, so $s_1 \neq s_2$.
- Trivial case 2: $s_1 \neq z$ and $s_2 = z$. Then, $s_1 \neq z = s_2$, so $s_1 \neq s_2$.
- Trivial case 3: $s_1 = z$ and $s_2 \neq z$. Then, $s_1 = z \neq s_2$, so $s_1 \neq s_2$.

At this point we are left with the non-trivial case: $z \neq y$, $s_1 \neq z$ and $s_2 \neq z$. Proceed by contradiction assuming the opposite $s_1 = s_2$. Because $\mathcal{R}$ fulfils the

convergence property, we know $y \notin \mathcal{R}(x, z)$ and $x \notin \mathcal{R}(z, y)$, so $s_1 \neq y$ and $s_2 \neq x$. Notice, however, that it is possible to have $s_1 = x$ and $s_2 = y$. But the assumption $s_1 = s_2$ leads to the following contradictions: $x = s_1 = s_2 \neq x$ and $y = s_2 = s_1 \neq y$. Therefore, $s_1 \neq s_2$ must hold. □

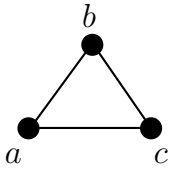
Hence the answer to the first part of Question 5.4 is: yes, strict size-monotonic recombination P-structures fulfil the inbreeding properties of geometric crossovers.

To finally prove the insufficiency of inbreeding properties, the second part of Question 5.4 asks for an example of a strict size-monotonic recombination P-structure that is a non-geometric crossover. Example 5.4 shows a strict size-monotonic recombination P-structure (X, OP) whose operator OP is not a geometric crossover on the specific metric space defined by a complete graph K_3 in Figure 5.7 below. However, that does not prove OP is a non-geometric crossover (i.e. $\text{OP} \in \overline{\mathcal{GX}}$): one has to prove that is not geometric for every metric, not just the metric of K_3. In that sense, Question 5.4 remains partially answered, and it is unclear how it may be fully answered. Let us explain then why it is unclear and why Theorem 5.16 is significant.

Example 5.4 (**Strict size-monotonic recomb. P-structure**)**.** Let the three-element set $X = \{a, b, c\}$. Define the binary operator $\text{OP} : X \times X \to \mathcal{P}(X)$ as:

$$\begin{aligned}
&\text{OP}(a,a) = \{a\};\\
&\text{OP}(b,b) = \{b\}; \quad \text{OP}(a,b) = \text{OP}(b,a) = \{a,b\};\\
&\text{OP}(c,c) = \{c\}; \quad \text{OP}(b,c) = \text{OP}(c,b) = \{b,c\}; \quad \text{OP}(a,c) = \text{OP}(c,a) = \{a,b,c\}.
\end{aligned}$$

Then, (X, OP) is a strict size-monotonic recombination P-structure by construction.



$$\begin{aligned}
&\text{OP}(a,a) = \{a\}\\
&\text{OP}(b,b) = \{b\}\\
&\text{OP}(c,c) = \{c\}\\
\hline
&\text{OP}(a,b) = \text{OP}(b,a) = \{a,b\}\\
&\text{OP}(b,c) = \text{OP}(c,b) = \{b,c\}\\
\hline
&\text{OP}(a,c) = \text{OP}(c,a) = \{a,b,c\}
\end{aligned}$$

Figure 5.7. A binary operator OP ('right') defined on the vertex set $X = \{a, b, c\}$ of a complete graph K_3 ('left'), such that (X, OP) is a strict size-monotonic recombination P-structure by construction, but OP is not a geometric crossover on the metric space (X, d) defined by K_3 since $\{a, b, c\} = \text{OP}(a, c) \not\subseteq [a, c]_d = \{a, c\}$ for parents $a, c \in X$.

The significance of Theorem 5.16 is proving the existence of a subclass of recombination P-structures ($\mathcal{RP}$-strictsize) that potentially proves the inbreeding properties

cannot guarantee if a crossover is geometric. This matters because if they are insufficient, then classifying crossovers as (non-)geometric becomes more complicated: the known alternative to the inbreeding properties is to verify Definition 3.3 by guessing distances, which is difficult for unfamiliar search spaces, or trial and error, which is impractical. Although Theorem 5.16 does not completely answer Questions 5.3–5.4, it is an starting point for two reasons. First, strict size-monotonic recombination P-structures $\mathcal{RP}$-strictsize, unlike geometric recombination P-structures, can contain non-geometric crossovers. Secondly, the axioms of $\mathcal{RP}$-strictsize (and any other recombination P-structure) do not involve metrics similar to the inbreeding properties. So it remains possible that $\mathcal{RP}$-strictsize or further subclasses thereof contain non-geometric crossovers (thus proving insufficiency); in that case, the axioms of such subclasses would precisely characterise those non-geometric crossovers.

The reason why it is unclear how to fully answer Questions 5.3–5.4 is the following problem: how do we prove that crossovers fulfilling the inbreeding properties, like strict size-monotonic recombination P-structures (see Example 5.4), are non-geometric if the only known viable test of non-geometricity relies on not satisfying the inbreeding properties? That is, proving non-geometricity without using the inbreeding properties. Three alternatives are:

(a) To propose new inbreeding properties, which leaves us with the original problem: prove again that such new properties are sufficient conditions.

(b) To prove that Definition 3.3 of geometric crossovers follows from the inbreeding properties (Propositions 3.2–3.4).

(c) To prove for every possible crossover and search space that, whenever it fulfils the inbreeding properties, it is geometric for some metric.

Except for option (b), options (a) and (c) are clearly not feasible. At this point, based on Theorem 5.16 and the existence of recombination P-structures that are non-geometric crossovers (Theorem 5.15), it may well be conjectured that the inbreeding properties are insufficient subject to further research on the relationship between non-geometric recombination P-structures and the inbreeding properties.

Conjecture 5.1. The inbreeding properties are insufficient conditions to guarantee if any given crossover is geometric for at least one metric space.

This section has not determined if the inbreeding properties are sufficient or not, but by exploiting the crossover classification (Section 5.4) it has provided formal evidence to support Conjecture 5.1, clarified its major difficulties, and found a new relevant subclass of recombination P-structures that may help prove it.

5.6 Conclusion

Carefully reconsidering the strengths and weaknesses of the GF and ELT, this chapter opens up a path to understand (across problems and representations) if and how the search space structure induced by crossovers influences crossover design as well as population behaviour or performance in EAs.

My original contribution is a crossover classification based on geometric crossovers and recombination P-structures. A classification that the GF and ELT never developed before or aimed to do so, as they focus exclusively on geometric crossovers or recombination P-structures not both. This classification reveals geometric recombination P-structures are a crossover subclass of both geometric crossovers and recombination P-structures, thereby justifying that a general unified theory of the GF and ELT is theoretically possible and not futile (Sections 5.3 and 5.4). Moreover, the crossover classification is not just a formal system to axiomatically compare geometric versus non-geometric crossovers with respect to their support structure, but also a promising means to investigate crossover design or EAs' behaviour in GF and ELT more deeply as discussed in Section 5.4.6. In fact, it became clear that principled design of geometric crossovers is possible and an integral part of ELT, which was unknown to the GF and especially to ELT as the latter never addressed crossover design at all.

This unified approach helped to identify and tackle other questions and misconceptions around crossover, relevant to the GF and ELT, that otherwise would have remained open: confusing geometricity and high-locality of crossovers (Section 5.5.1), and whether the inbreeding properties are sufficient to guarantee geometricity (Section 5.5.2). Thus classifying crossovers for its own sake is not the aim of the classification here presented, but rather to formally understand the axiomatic properties inherent to crossovers across problems and representations. This chapter accomplished it for both geometric crossovers and recombination P-structures comprehensively and systematically, which in no way previous work in the GF (Chapter 3) nor ELT (Chapter 4) could do independently from each other.

Part II

Evolutionary Search

Abstract

This second part develops upon Part I to unify the general models of evolutionary search proposed by the GF and ELT, revised next in background Chapters 6 and 7. EAs with crossover but no mutation are generalised in the GF across metric spaces as geometric-crossover EAs: any EA based on geometric crossover parametrised by an unspecified metric distance. Interestingly, their abstract population behaviour can be formalised solely in terms of geodesic convexity associated with metric spaces: an abstract form of (geodesically) convex evolutionary search, independent of problems and representations, intuitively described as a sequence of nested convex shapes. By contrast, ELT conceives two different approaches to model evolutionary search depending on the kind of search space. For mutation search spaces, ELT defines random walks on connected simple graphs, based on the neighbourhood induced by mutation, or more specifically: time-homogeneous, reversible, and irreducible, finite Markov chains. For crossover search spaces, or recombination P-structures, ELT defines a generalised random walk parametrised by a certain probability distribution, which is a form of headless-chicken crossover random walk. The associated finite Markov chain depends on the probability distribution used and recombination P-structure (hypergraph).

The contribution of this second part is presented in Chapter 8, adopting the GF's approach as a basis because recombination P-structure random walks do not model population-based EAs. The focus is on the two-fold research question (2) from the literature review in Chapter 2: what class of EAs doing abstract convex evolutionary search is shared by the GF and ELT? Can it use mutation operators? To address these questions, Chapter 8 extends the crossover classification laid in Chapter 5 with a new crossover class described by finite interval operators, not necessarily based on metric spaces nor geodesic convexities, which includes all (geometric or non-geometric) recombination P-structures but also crossovers similar to the macro-mutations present in headless-chicken crossover random walks. This new crossover class leads to a corresponding generalised class of EAs (possibly using mutation), called formal interval EAs, which undertake a generalised form of the abstract convex evolutionary search carried by geometric-crossover EAs. As a particular case of formal interval EAs, Chapter 8 shows that the subclass of geometric recombination P-structure EAs, without mutation, shared by GF and ELT, provably do abstract (geodesically) convex evolutionary search. Overall, Chapter 8 develops a qualitative framework that simplifies and systematises the task of axiomatically comparing how different crossovers classes affect the abstract search behaviour of EAs using them.

Chapter 6

Evolutionary Search in the Geometric Framework

This chapter revises background material for subsequent chapters. It summarises and clarifies key ideas in the GF about abstract convex search of EAs using geometric operators (Chapter 3). This chapter principally supports the contributions in Chapter 8 generalising such abstract search behaviour to EAs based on recombination P-structures introduced in Chapter 4.

6.1 From EAs to Geometric-crossover EAs

The fundamental mechanism behind the numerous and diverse kinds of EAs that exist fits the following pseudo-code [4].

```
1 Generate an initial population P
2 Evaluate P
3 while termination criteria is unsatisfied do
4 |   Select a subset P_1 from P as parents
5 |   Recombine P_1 to produce offspring P_2
6 |   Mutate P_2
7 |   Evaluate P_2
8 |   Replace P with P_2
9 return P
```

Algorithm 6.1. Pseudo-code of a conventional evolutionary algorithm.

Algorithm 6.1 is a simplified and general description of an EA that has been stripped of specific representation of solutions, fitness function(s), mutation and crossover rates, among other factors. Each of these factors involves a choice that may lead to completely different specific EAs. To provide a common and coherent

basis across EAs, the GF proposed by Moraglio [100] defines a general model of EAs, which is the subject for this chapter.

Distance is the fundamental notion in the GF to abstract search spaces as metric spaces, where generalisations of mutation and crossover operators used in practice are defined by metric balls and metric segments (Chapter 3). Such geometric operators (Definitions 3.3–3.4) are regarded as parametrised by a metric (i.e. distance function). Similarly, one may regard EAs using geometric operators as parametrised by a metric and treat them as a formal object relying only on the metric axioms. Doing so, while not specifying any metric, turns Algorithm 6.1 into a formal EA independent of the underlying search space structure and solutions' representation. The GF names it *formal metric evolutionary algorithm* (Definition 6.1) and *abstract evolutionary search* its behaviour (Definition 6.2) to emphasise that they are formal entities depending only on the axioms of metric spaces.

Definition 6.1 (**Formal metric EA** [100, 101])**.** A *formal metric evolutionary algorithm* is any EA using geometric operators on a fixed but unspecified metric space (X, d) where X is the set of all candidate solutions and the metric d imposes a structure on X.

Definition 6.2 (**Abstract evolutionary search** [100, 101])**.** *Abstract evolutionary search* is the abstract behaviour of a formal metric EA across all metric spaces. That is, the behaviour common to all EAs obtained from the formal metric EA by specifying a set of solutions and a metric.

Based on Algorithm 6.1, the key components of a formal metric EA at a population-level are thus [100, 101]:

- A *population* of individuals where each individual represents a candidate solution and may occur multiple times in the population (i.e. a multi-set of individuals).
- *Fitness* and *distance functions* are unspecified but fixed[1]. The following population operators may internally use these functions.
- *Selection*: an operator that accepts an input population and outputs another population where some input individuals may have been eliminated, or their multiplicity may have increased or decreased, but the underlying set of the output population is a subset of the input population.

[1]Time-varying fitness functions are not excluded, so 'fixed' should not be confused with 'static'.

- *Recombination/Crossover*: an operator that accepts an input population and outputs another population by applying any geometric crossover operator any desired number of times to pairs of individuals in the input population.
- *Mutation*: an operator that accepts an input population and outputs another population by applying any geometric mutation operator any desired number of times to any individual in the input population.
- *Replacement*: an operator that accepts two input populations P_1 and P_2 and outputs a population P_3 by applying the union[2] operation of multi-sets to P_1 and P_2, and then applying *selection* to the population that results from their union so that $P_3 \subseteq P_1 \cup P_2$ (with respect to their underlying sets).

There are no restrictions on whether some of these population operators are deterministic or probabilistic. In particular, any probability distribution may be imposed over offspring obtained by geometric operators as long as their formal Definitions 3.3–3.4 are respected. Moreover, any of the population operators may degenerate to the case where the output population equals the input population (e.g. a probabilistic mutation operator may not mutate any individual at all).

Besides, if mutation is not used, then one obtains a class of formal metric EAs called *geometric-crossover* EAs[3] (Definition 6.3) in which crossover is the only operator introducing evolutionary change in populations (i.e. new individuals).

Definition 6.3 (**Geometric-crossover EA** [100, 101])**.** A *geometric-crossover evolutionary algorithm* is any formal metric EA that does not mutate populations.

Geometric-crossover EAs exhibit a very distinctive abstract behaviour where populations through generations become more 'localised' in a region of the search space delimited by the initial population. Such behaviour can be characterised using abstract convexity theory [146], hence its name *abstract convex evolutionary search* [100, 101]. Section 6.2 reviews some basic notions on convexity prior to formalising abstract convex evolutionary search in Section 6.3.

6.2 Abstract Convexity Preliminaries

To understand the behaviour of geometric-crossover EAs, it is key to understand how offspring relate to parents at a population's level in terms of geometric crossovers.

[2]The union of multi-sets takes the maximum of the occurrences for given multi-sets. Alternatively, the sum operation may be used, which takes the sum of the occurrences for given multi-sets.
[3]A software implementation of a geometric-crossover EA used in [104] is available at [102].

For that, Section 6.2.3 introduces an abstract notion of convexity used later in Section 6.3 to formalise such behaviour and offspring-parent relationship. But first let us have a look at convexity in Euclidean (Section 6.2.1) and Hamming metric spaces (Section 6.2.2) to gain some intuition.

6.2.1 Geodesic Convexity in Euclidean Metric Spaces

In Euclidean spaces, and other vector spaces, line segments and convex combinations [12] capture the intuitive idea of offspring lying 'between' two parents and, respectively, a set of parents or population.

Consider a given set of parents $A = \{a_1, \ldots, a_5\} \subset \mathbb{R}^2$ as illustrated in Figure 6.1. Then, every point $a' \stackrel{\text{def}}{=} \theta_1 a_1 + \cdots + \theta_5 a_5$, such that $\theta_i \geq 0$ and $\sum_i \theta_i = 1$, is a *convex combination* of the points in A. Here, a' may be considered an offspring resulting from a weighted combination of the parents $a_1, \ldots, a_5$ using weights $\theta_1, \ldots, \theta_5$. The set of all those possible offspring, namely all convex combinations of A, is its *convex hull* $co(A) \stackrel{\text{def}}{=} \{\sum_i \theta_i a_i \mid a_i \in A,\ \theta_i \geq 0,\ \sum_i \theta_i = 1\}$. Equivalently, $co(A)$ is the 'smallest' convex set including A. Restricting A to two parents $co(\{a_1, a_2\})$ precisely results in the *line segment* $\theta a_1 + (1-\theta)a_2$ between the two points $a_1, a_2 \in \mathbb{R}^2$ with $\theta \in [0, 1]$. Therefore, it is clear that for all $x, y \in A : [x, y]_{d_2} = co(\{x, y\}) \subseteq co(A)$ under the Euclidean metric d_2. So a geometric crossover in Euclidean space produces offspring in the line segment between two parents, and at a population's level within its convex hull.

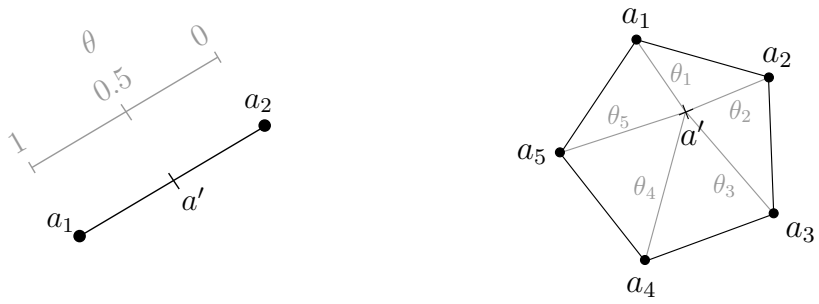


Figure 6.1. A line segment ('left') and a convex combination ('right') in the Euclidean metric space $(\mathbb{R}^2, d_2)$.

6.2.2 Geodesic Convexity in Hamming Metric Spaces

For Hamming metric spaces, Hamming segments and Holland's schemas [65] capture the intuitive idea of offspring lying 'between' two parents or a set of them.

A *schema* is a string of length n using a q-ary alphabet for $q \geq 2$, like Hamming sequences, but with an extra 'wildcard' symbol $*$ [65]. The set of all such schemas

is $\{0, 1, \ldots, q-1, *\}^n$, and a wildcard represents any of $\{0, 1, \ldots, q-1\}$. So `2*1*` is a schema for $n = 4$ and $q = 3$: it represents all Hamming sequences formed by substituting all wildcards by any of the other symbols (i.e. `0`, `1` or `2`), and fixing `2` and `1` at the first and third leftmost positions respectively. Not substituting all wildcards produces a subspace of the original schema, for instance: `201*` $\subseteq$ `2*1*`.

Every schema defines a convex set (compare Proposition A.1 in [98] and the definition of convexity in [146, ch. 1]). This is also justified by the fact that the convexity structure of a Hamming graph is the product convexity of its factors [146], namely a n-dimensional q-ary Hamming graph $H(n, q)$ is formed by the graph Cartesian product of n copies of the complete graph K_q (which is convex). Consequently, the schema of *lowest order*, namely having fewest $*$ symbols, that matches a set of Hamming sequences is the convex hull of that set (i.e. 'smallest' convex set that includes it). For instance, let $n = 3$, $q = 2$ and a population P of individuals `001`, `101`, `000`

$$P = \begin{Bmatrix} (0 & 0 & 1), \\ (1 & 0 & 1), \\ (0 & 0 & 0) \end{Bmatrix} .$$

Then, `*0*` is the lowest-order schema that matches P. So the convex hull of P is $co(\{\texttt{001}, \texttt{101}, \texttt{000}\}) = \{\texttt{000}, \texttt{001}, \texttt{100}, \texttt{101}\} \equiv \texttt{*0*}$. The schema `***` also matches P, except it is not of lowest order. If P is limited to two parents, then its lowest-order schema also corresponds to offspring under uniform crossover (Definition 3.5):

$$\textsc{Uniform}(\texttt{101}, \texttt{001}) = \{\texttt{101}, \texttt{001}\} = \overbrace{\{\texttt{1}, \texttt{0}\}}^{\texttt{*}} \times \overbrace{\{\texttt{0}, \texttt{0}\}}^{\texttt{0}} \times \overbrace{\{\texttt{1}, \texttt{1}\}}^{\texttt{1}} \equiv \texttt{*01}$$

or equally the Hamming segment between parents $[\texttt{101}, \texttt{001}]_{d_\mathrm{H}}$ for the Hamming metric d_H. In other words, for all $x, y \in P : [x, y]_{d_\mathrm{H}} = co(\{x, y\}) \subseteq co(P)$. So a geometric crossover in Hamming space produces offspring in the Hamming segment between two parents or within a schema (i.e. convex hull) of a set of parents.

6.2.3 Geodesic Convexity in General Metric Spaces

Geometric crossovers are well defined on every metric space. Therefore, it is appropriate to introduce a convexity that generalises those of Euclidean and Hamming metric spaces (Section 6.2.1–6.2.2) to any metric space. A natural choice is the *geodesic convexity* associated with metric spaces (Definition 6.4) since geometric crossovers are defined by geodesic intervals (Definition 3.2). To each geodesic convex space there corresponds a unique *geodesic convex hull* (Definition 6.5).

Definition 6.4 (**Geodesic convex space** [146])**.** Let (X, d) be any metric space. A point $z \in X$ is *geodesically between* two points $x, y \in X$, if and only if $z \in [x, y]_d$. A subset $C \subseteq X$ is *geodesically convex*, if and only if all points geodesically between any two points x and y in C are also in C; that is, if $\forall x, y \in C : [x, y]_d \subseteq C$. The family $\mathcal{C}$ of all geodesically convex subsets C in X is the *geodesic convexity* of (X, d), and the pair $(X, \mathcal{C})$ is a *geodesic convex space*.

Definition 6.5 (**Geodesic convex hull** [146])**.** Let $(X, \mathcal{C})$ be the geodesic convex space of any metric space (X, d). The *geodesic convex hull closure operator*, or *geodesic convex hull* for short, of a set A in X is the intersection of all geodesically convex supersets $C \in \mathcal{C}$ of A, or equally the smallest geodesically convex set including A, under the metric d. Formally, it is denoted $co : \mathcal{P}(X) \to \mathcal{P}(X)$ and defined on any $A \in \mathcal{P}(X)$ by $co(A) \stackrel{\text{def}}{=} \bigcap\{C \mid A \subseteq C \in \mathcal{C}\}$, $\mathcal{P}(X)$ being the power set of X.

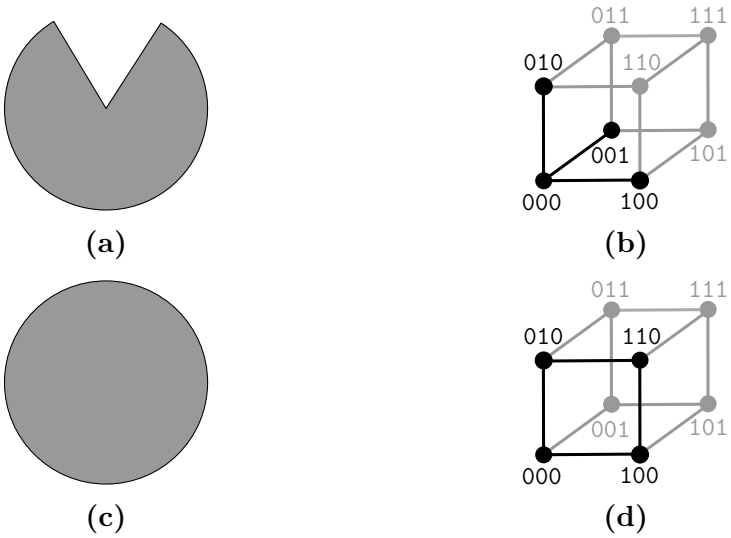


Figure 6.2. Examples of geodesically non-convex (Figures 6.2a–6.2b) and convex sets (Figures 6.2c–6.2d) in two-dimensional Euclidean metric space $(\mathbb{R}^2, d_2)$ and three-dimensional binary Hamming metric space $(\mathcal{H}_2^3, d_{\mathrm{H}})$.

Observe that Definitions 6.4–6.5 are parametrised by a metric. So by substituting a specific metric such as the Euclidean or Hamming metric one obtains their associated specific geodesic convexities (Sections 6.2.1–6.2.2) and specific notions of geodesic (non-)convex sets (Figure 6.2), likewise for other metrics.

However, that all geodesic convexities share Definitions 6.4–6.5 does not imply necessarily that all properties of a specific geodesic convexity are true for another geodesic convexity [146]. For instance, unlike Euclidean balls (Figure 6.2c), Hamming balls are not geodesically convex in general: the Hamming segment $[\mathtt{100}, \mathtt{001}]_{d_{\mathrm{H}}}$ is not completely included in the Hamming ball of Figure 6.2b.

Furthermore, even if metric segments and geodesic convex hulls sometimes coincide[4], as seen in Sections 6.2.1–6.2.2, in general that is false because $\forall x, y \in S \subseteq X$: $[x, y]_d \subseteq co(\{x, y\}) \subseteq co(S)$ for every metric space (X, d) [146]. Figure 6.3 shows an example where not all metric segments are geodesic convex hulls.

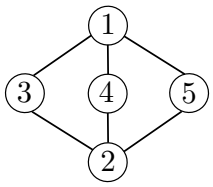


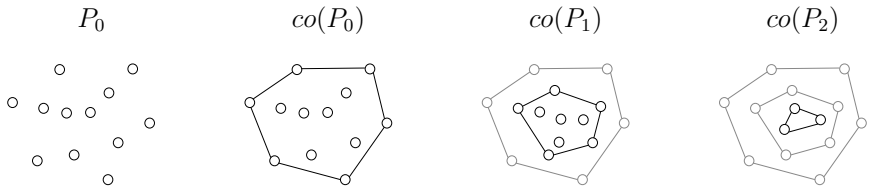
Figure 6.3. A $K_{2,3}$ bipartite graph where metric segments (i.e. shortest-paths) do not always equal convex hulls: $[3, 5]_d = \{3, 1, 2, 5\} \subset \{3, 1, 2, \underline{4}, 5\} = co(\{3, 5\})$, d being the shortest-path metric induced by $K_{2,3}$.

6.3 Behaviour of Geometric-crossover EAs: Abstract Convex Evolutionary Search

The GF states that the core behaviour of EAs, specifically geometric-crossover EAs (Definition 6.3), can be described axiomatically across representations and problems; and, ultimately, the differences between EAs stem from the solutions' representation and search operators [100, 101]. To support this claim, the GF proved that all geometric-crossover EAs exhibit the same behaviour: *abstract (geodesically) convex evolutionary search* (Proposition 6.1, see Theorem 15.4.3 in [100]) regardless of the representation of solutions, problem (fitness function), geometric crossover with any probability distribution, selection and replacement population operators, and whether the population size varies or not through generations. Figure 6.4 shows such abstract convex behaviour for an hypothetical geometric-crossover EA in Euclidean and Hamming metric spaces based on their associated convexities (Section 6.2).

Proposition 6.1 (**Abstract** [**geodesically**] **convex evolutionary search** [100, 101])**.** Let (X, d) be any metric space and $P_t \subseteq X$ a population at generation $t \geq 0$, with P_0 being the initial population. Then, for any geometric-crossover EA repeating the cycle of population operators selection, crossover and replacement: $co(P_0) \supseteq co(P_1) \supseteq \cdots \supseteq co(P_t) \supseteq co(P_{t+1})$.

[4]For geodesic convex spaces that coincide with *convex geometries* [41], $[x, y]_d \subseteq co(\{x, y\})$ becomes $[x, y]_d = co(\{x, y\})$ if x and y are *extreme points* of $[x, y]_d$ (i.e. $[x, y]_d \setminus \{x, y\}$ is also geodesically convex). That is because in convex geometries every (geodesically) convex set C is the convex hull of the extreme points of C. Graphs with such property are Ptolemaic graphs [41, 112].



(a)

P_0	$co(P_0)$	$co(P_1)$	$co(P_2)$
$\left\{\begin{matrix}(0\ 0\ 0\ 1\ 1),\\(0\ 1\ 0\ 0\ 1),\\(1\ 0\ 0\ 0\ 1),\\(1\ 1\ 0\ 1\ 1)\end{matrix}\right\}$	`**0*1`	`0*0*1`	`010*1`

(b)

Figure 6.4. Abstract (geodesically) convex evolutionary search during two generations for an initial population P_0 in: two-dimensional Euclidean metric space $(\mathbb{R}^2, d_2)$ (Figure 6.4a) and five-dimensional binary Hamming metric space $(\mathcal{H}_2^5, d_{\mathrm{H}})$ (Figure 6.4b).

Thus abstract convex evolutionary search[5] produces a nested inclusion chain of populations' geodesic convex hulls that is restricted to the geodesic convex hull of the initial population $co(P_0)$: all individuals outside $co(P_0)$ will not be generated in future generations. For Euclidean metric spaces (Figure 6.4a), each individual at each generation is a convex combination of the parent population. For Hamming metric spaces (Figure 6.4b), each individual at each generation belongs to the schema of the parent population. When a population operator of a geometric-crossover EA (or formal metric EA in general) produces offspring within the geodesic convex hull of the parent population, it is called a *convex population operator*[6] (Definition 6.6).

Definition 6.6 (**Convex population operator** [100, 101])**.** Let (X, d) be any metric space and multi-sets $P, P' \in \mathbb{N}^X$ be arbitrary populations such that $P' \subseteq co(P)$. Denote $s, c : \mathbb{N}^X \to \mathbb{N}^X$ and $r : \mathbb{N}^X \times \mathbb{N}^X \to \mathbb{N}^X$ to be selection s, crossover c and replacement r population operators. Then, s, c and r are *convex population operators*, if $s(P) \subseteq co(P)$, $c(P) \subseteq co(P)$ and $r(P, P') \subseteq co(P)$ respectively, in terms of the underlying sets of P, P'.

Precisely, the reason why geometric-crossover EAs do abstract convex evolutionary search is that selection, crossover and replacement (Section 6.1) are convex

[5]The term 'abstract convex evolutionary search' is used as in 'abstract geodesically convex'. This thesis makes it explicit when discussing non-geodesic convexities in later chapters.

[6]The original name 'convex operator' may be mistakenly confused with convex hull operators. To avoid confusion, this thesis renames it as 'convex population operator' to emphasise that it refers to the population operators of EAs, namely selection, crossover, etc.

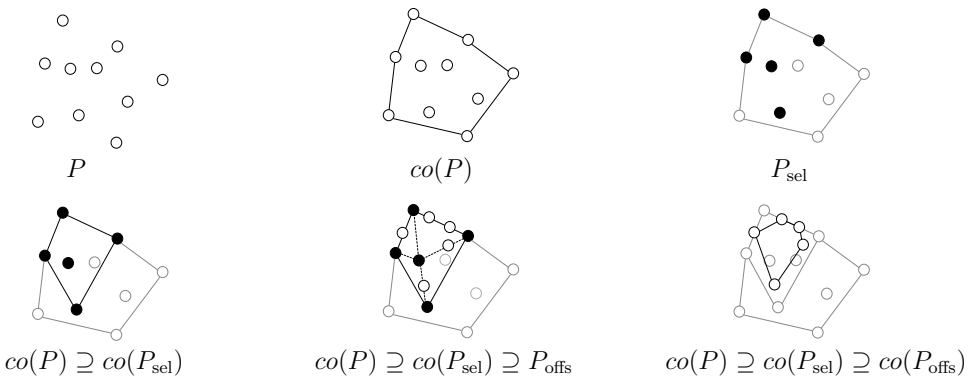


Figure 6.5. Abstract (geodesically) convex evolutionary search in two-dimensional Euclidean metric space $(\mathbb{R}^2, d_2)$: relationship between parent P, mating pool after selection P_{sel} and offspring P_{offs} populations, and their respective convex hulls $co(P)$, $co(P_{\text{sel}})$ and $co(P_{\text{offs}})$.

population operators (Proposition 6.2, see Theorem 15.4.2 in [100]). For example, Figure 6.5 illustrates in Euclidean metric space that selecting a mating pool of parents P_{sel} from a given population P and then recombining them (using some unspecified geometric crossover) produces an offspring population P_{offs} such that $co(P) \supseteq co(P_{\text{sel}}) \supseteq co(P_{\text{offs}})$.

Proposition 6.2 ([100, 101])**.** For any geometric-crossover EA, selection, crossover and replacement are convex population operators.

6.3.1 Abstract Non-convex Evolutionary Search

In contrast with geometric-crossover EAs (Definition 6.3), formal metric EAs (Definition 6.1) can use mutation. As a result, their abstract evolutionary behaviour (Definition 6.2) need not be abstract convex evolutionary search (Proposition 6.1) because mutation is provably not a convex population operator (Definition 6.6) [100, 101], except for degenerate cases where a probabilistic mutation operator mutates no individual.

Put differently, mutation may produce offspring outside the geodesic convex hull of a parent population, breaking thus the nested inclusion chain that characterises abstract convex evolutionary search. For example, recall the nested chain of schemas in Figure 6.4b for the first two generations of an initial population P_0

$$\overbrace{\texttt{**0*1}}^{co(P_0)} \supseteq \overbrace{\texttt{0*0*1}}^{co(P_1)} \supseteq \overbrace{\texttt{010*1}}^{co(P_2)} .$$

Suppose `00011` is an individual of P_1, so $\texttt{00011} \in co(P_1)$. Then, applying single-bit

flip mutation to the rightmost bit `0` (i.e. `00`$\underline{\texttt{0}}$`11`) produces `00111` $\notin co(P_0) \equiv$ `**0*1`, thus breaking the nested chain.

Arguably, if the disruptive effects of mutation are somehow 'restricted', then an approximated convex evolutionary search may be possible [100]. However, its formalisation remains as future work [104] since it requires extending the GF from metric spaces to metric measure spaces (incorporating probabilities besides distances) [101]. It is out of the scope of this thesis to address such extension.

Chapter 7

Evolutionary Search in Elementary Landscapes Theory

This chapter revises background material for subsequent chapters. It summarises and clarifies key ideas in ELT about two degenerate forms of evolutionary search, random walks and crossover random walks, associated with mutation search spaces (defined by neighbourhoods) recombination search spaces (defined by recombination P-structures) introduced in Chapter 4. This chapter principally supports the contributions in Chapter 8 providing ELT with a proper and more general EA model using crossovers defined by recombination P-structures.

7.1 Evolutionary Search Rooted in Search Spaces

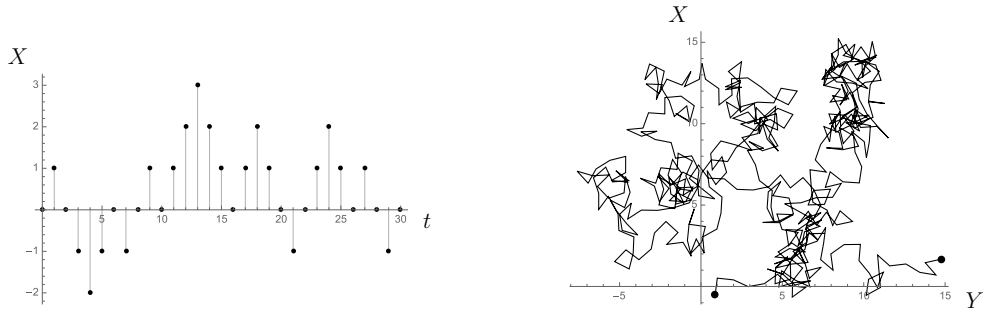
Evolution by means of natural selection [27], and its application in EC [4], inspired research in ELT [136, 140, 150] towards a mathematical theory for the analysis of complex adaptive systems (e.g. ensembles of particles, organisms, etc.) [65], focusing on the algebraic structure of search spaces and fitness landscapes more than the population behaviour of EAs. The behaviour of populations undergoing selection, recombination and mutation has been studied in connection with the so-called elementary landscapes via dynamical systems [138], but departing from the ELT graph-theoretical approach based on search space structures (see Chapter 10).

ELT distinguishes between search spaces induced by mutation and those induced by recombination (Chapter 4). Although ELT is a theory about landscapes, such distinction leads to two separate search strategies for each kind of search space: random walks for mutation search spaces (Section 7.2) and for recombination search spaces a form of random walk that 'fakes' the action of crossovers (Section 7.3).

7.2 Random Walk

Random walks [19, 110] are stochastic processes that consist of a sequence of events occurring at discrete time steps and where future events depend on the present (subject to a random perturbation) but are independent from the past. That is, random walks have the *Markov property* for which they are named also *discrete-time Markov processes*. Figure 7.1 illustrates two examples of random walks where events are described by random variables in discrete and continuous spaces.

Particularly, ELT considers discrete-state discrete-time Markov processes (i.e. *finite Markov chains*) where random variables take values from a finite set, called the set of *states*, such as a graph's vertex set. Random walks on graphs or mutation search spaces are then finite Markov chains as Section 7.2.1 explains.



(a) **(b)**

Figure 7.1. A one-dimensional random walk plotted against time t for an integer-valued random variable X (Figure 7.1a). A two-dimensional random walk, after some unspecified time, with real-valued random variables X and Y whose values have been interpolated by lines to appreciate the random walk's trajectory (Figure 7.1b).

7.2.1 Finite Markov Chain: Definition and Example

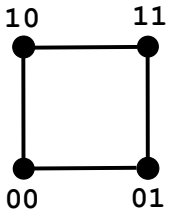
In ELT, a mutation search space is a pair (X, N). X being a finite set of all candidate solutions and $N : X \to \mathcal{P}(X)$ a neighbourhood function describing the support function of a mutation operator, which imposes a structure on X by mapping $x \in X$ to a subset $N(x) \in \mathcal{P}(X)$ of neighbour solutions. ELT represents mutation search spaces as graphs since N determines an adjacency matrix (see Section 4.1). Thus ELT defines the following finite Markov chain for a connected graph G with vertex set $V = \{v_1, \ldots, v_n\}$ where $V = X$ and the edge set E is given by N. The *set of states* of the Markov chain is the vertex set V, and the *transition probabilities* p_{v_i,v_j} from state v_i to state v_j are given by its *transition probability matrix* $\mathbf{P}$ [136, 137]

$$\mathbf{P} \stackrel{\text{def}}{=} \mathbf{A}\mathbf{D}^{-1} \tag{7.1}$$

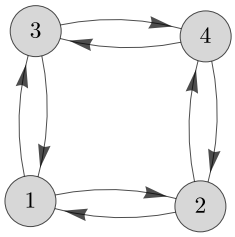
where the adjacency $\mathbf{A}$ and (diagonal) vertex-degree[1] $\mathbf{D}$ matrices of the given graph G have entries

$$a_{v_i,v_j} \stackrel{\text{def}}{=} \begin{cases} 1, & \text{if } \{v_i, v_j\} \in E, \\ 0, & \text{otherwise,} \end{cases} \qquad d_{v_i,v_j} \stackrel{\text{def}}{=} \begin{cases} |N(v_i)|\,, & \text{if } v_i = v_j, \\ 0, & \text{otherwise} \end{cases} \qquad \forall v_i, v_j \in V.$$

Then, the Markov chain on G is determined by the pair $(\mathbf{p}^{(0)}, \mathbf{P})$. The row vector $\mathbf{p}^{(0)}$ is the initial probability distribution over the set of states, and the corresponding probability distribution after t time steps is given by the t-th power of $\mathbf{P}$: $\mathbf{p}^{(t)} = \mathbf{p}^{(0)}\,\mathbf{P}^t$. So $\mathbf{p}^{(0)}$ describes an initial solution of a random walk on G and $\mathbf{p}^{(t)}$ a solution after t steps of such random walk.



(a)



(b)

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

(c)

$$\mathbf{D} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

(d)

$$\mathbf{P} = \mathbf{A}\mathbf{D}^{-1} = \begin{bmatrix} 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix}$$

(e)

Figure 7.2. State transition diagram (Figure 7.2b), whose labels $1, \ldots, 4$ correspond to $\mathtt{00}, \ldots, \mathtt{11}$, and transition probability matrix $\mathbf{P}$ (Figure 7.2e) of a Markov chain for the mutation search space $(\mathcal{H}_2^2, \textsc{Bitflip}(1))$ induced by single-bit flip neighbourhood $\textsc{Bitflip}(1)$ on two-dimensional binary Hamming sequences $\mathcal{H}_2^2$. Figure 7.2a shows its associated Hamming graph with adjacency $\mathbf{A}$ and vertex-degree $\mathbf{D}$ matrices (Figures 7.2c–7.2d).

For example, consider the multiple-bit flip neighbourhood (Definition 4.1), specifically the single-bit flip neighbourhood $\textsc{Bitflip}(1)$ that outputs all Hamming sequences one bit flip away from a given one.

The mutation search space induced by $\textsc{Bitflip}(1)$ on two-dimensional binary Hamming sequences is the Hamming graph in Figure 7.2a. A Markov chain for such Hamming graph has states the vertices $v_1 = \mathtt{00}, \ldots, v_4 = \mathtt{11}$. The transition probability matrix $\mathbf{P}$ for such states (Figure 7.2e) follows using Equation 7.1 from the adjacency and vertex-degree matrices (Figures 7.2c–7.2d) of the Hamming graph.

[1]The degree of a vertex is the number of vertices adjacent to it.

Hence $\mathbf{P}$ alongside any initial distribution over the states determines a Markov chain, with state diagram in Figure 7.2b, on the mutation search space induced by the single-bit flip neighbourhood BITFLIP(1). Moreover, the transition probabilities reflect the neighbourhood structure of BITFLIP(1): when $p_{v_i,v_j} > 0$, the vertex v_j is one bit flip away from vertex v_i in the Hamming graph, namely $v_j \in \text{BITFLIP}(1)(v_i)$. Note that regardless of the initial distribution and time step of the Markov chain:

- The transition probabilities are constant.
- Each transition is equiprobable in both directions. For instance, moving from state `00` to state `01` occurs with probability $\frac{1}{2}$, which equals that of moving from state `01` to `00`.
- All four states are accessible from any state after some time. For instance, state `11` is not directly accessible from state `00` since $p_{\texttt{00},\texttt{11}} = 0$, which is in turn due to $\texttt{11} \notin \text{BITFLIP}(1)(\texttt{00})$. But state `11` is accessible from state `00` after two bit flips (i.e. two time steps) since $\texttt{01} \in \text{BITFLIP}(1)(\texttt{00})$, $\texttt{11} \in \text{BITFLIP}(1)(\texttt{01})$, and the corresponding transition probabilities $p_{\texttt{00},\texttt{01}}$ and $p_{\texttt{01},\texttt{11}}$ are not zero.

The next section clarifies why these three properties hold not just for the example in Figure 7.2 but every Markov chain as defined in ELT by Equation 7.1. Appendix A.5 includes a software implementation of the Markov chain in the previous example.

7.2.2 Finite Markov Chain: Properties

To understand the differences between the two major forms of random walk discussed in this chapter, it is necessary to remark some key aspects and assumptions made in ELT about the Markov chain presented in Section 7.2.1. Its transition probability matrix $\mathbf{P}$ in Equation 7.1 is a specific case of a more general Markov chain with transition probability matrix $\mathbf{P}^{(t)}$ at time step t, transition probabilities $p_{i,j}^{(t)}$, and set of states S that has the following characteristics [110]:

(a) $\mathbf{P}^{(t)}$ is a *non-negative* matrix because $p_{i,j}^{(t)} \in [0, 1]$ for every time step $t \in \mathbb{N}_0$ and states $i, j \in S$.

(b) $\mathbf{P}^{(t)}$ is a *stochastic* matrix because the entries of each row $p_{i,\cdot}^{(t)}$ add up to one: $\sum_{j \in S} p_{i,j}^{(t)} = 1$ for every time step $t \in \mathbb{N}_0$ and state $i \in S$.

(c) *Time-homogeneous* because the transition probabilities do not change over time: $\mathbf{P}^{(t_1)} = \mathbf{P}^{(t_2)}$ for every two time steps $t_1, t_2 \in \mathbb{N}_0$. For time-homogeneous Markov chains, $\mathbf{P}^{(t)}$ is simply written as $\mathbf{P}$.

(d) *Reversible* because $\mathbf{P}^{(t)}$ is symmetric: $p_{i,j}^{(t)} = p_{j,i}^{(t)}$ for every time step $t \in \mathbb{N}_0$ and states $i, j \in S$.

(e) *Irreducible* because it is possible to move from any state $i \in S$ to any other state $j \in S$ eventually.

A Markov chain need not fulfil properties (c–e), but ELT assumes that the Markov chain given by $\mathbf{P}$ in Equation 7.1 does. In fact, that is partly due to the usual assumptions[2] made about graphs in ELT [136]: connected (so there is a path between any two vertices), undirected (so the adjacency matrix is symmetric) and regular (so all vertices have the same number of neighbours, i.e. all diagonal entries of the vertex-degree matrix are equal). For example, if ELT allowed directed graphs, then $\mathbf{P}$ may not be reversible since directed graphs need not have symmetric adjacency matrices.

7.3 Headless-chicken Crossover Random Walk

EAs are a natural choice to model population dynamics under the effects of crossover [4, 52]. Generalisations of random walks involving populations and crossover also exist: crossover random walks [139, 140, 150]. Before Section 7.3.2 presents their formal definition, Section 7.3.1 introduces first what is a crossover random walk intuitively and what it has to do with 'headless chickens'.

7.3.1 Overview and the Headless Chicken Metaphor

Using generalised forms of crossover to move between states in random walks appears in ELT [139] with the introduction of certain interval structures called recombination P-structures seen in Chapter 4. Later, ELT formalised how distinct probability distributions over a set of solutions, from which parents are sampled for recombination and that may be fixed and independent from the current state of the search, affect such crossover random walks [140, 150].

The overall idea of a crossover random walk is the following. First, generate an initial candidate solution (the 'father'), then sample a second candidate solution (the 'mother') from a probability distribution over the set of solutions (possibly the entire search space), recombine both and randomly choose one of the offspring as the next father to repeat this cycle. In this manner, the sequence of fathers (or

[2]ELT is extensible to directed and non-regular graphs [5, 35], but this thesis does not consider them since ELT is not as well developed in such cases.

offspring) yields the sequence of states visited by the crossover random walk whose pseudo-code is Algorithm 7.1. Although any crossover may be used in principle, ELT focuses on those represented by recombination P-structures (see Section 7.3.2 for details), but the idea of crossover random walk is the same.

```
1 Generate an initial candidate solution x ∈ X
2 while termination criteria is unsatisfied do
3     Set the current state to x
4     Sample a new candidate solution y from a probability distribution over X
5     Recombine x and y
6     Choose at random an offspring z of x and y; discard the rest
7     Replace x with z
```

Algorithm 7.1. Pseudo-code of a crossover random walk.

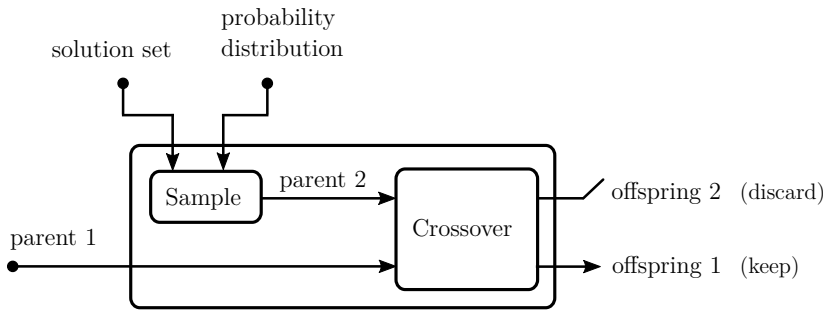


Figure 7.3. Diagram of a headless-chicken crossover (or macro-mutation).

This kind of random walk is originally known as *headless-chicken (HC) crossover* random walk [73]. Its name metaphorically alludes to the fact that a 'headless chicken' is not really a chicken, and likewise a 'crossover' where one of the parents is randomly chosen not from a population but a probability distribution over the search space (as in Algorithm 7.1) is not really a crossover operation. Indeed HC crossover[3] is deemed a generalised mutation (or macro-mutation) operation [73] since both mutation and HC crossover accept one input individual and randomly modify it to output one new individual as Figure 7.3 illustrates.

The idea of crossover in EAs is to combine solutions taken from a parent population to form an improving offspring population of solutions [4]. However, HC crossover random walks (Algorithm 7.1) differ from conventional EAs: no populations are used, new individuals are created by macro-mutations on a prescribed individual rather than by recombining parents in a population, and neither replacement nor selection mechanisms are enforced to successively produce fitter populations

[3]HC crossovers appear in the literature under other names like 'random-mate random-child' [66].

over generations. Thus HC crossover random walks are essentially random walks, or degenerate types of EAs, doing a form of random search.

7.3.2 Finite Markov Chain: Definition and Example

In a HC crossover (Figure 7.3) the first parent x is given and the second parent y is sampled from a probability distribution over a solution set S, then only one offspring z is kept and reused as first parent for the next recombination. This describes a transition from the current state x to the next state z in any HC crossover random walk, where states correspond to solutions in S. Probabilistically, the current and next states are represented by a random variable X at time steps t and $t+1$, namely $X^{(t)}$ and $X^{(t+1)}$, and the sampled parent by an independent random variable $Y^{(t)}$. Then, the transition probability at time step t is

$$\underbrace{\Pr\{X^{(t+1)} = z \mid X^{(t)} = x\}}_{\text{(a)}} \stackrel{\text{def}}{=} \sum_{y \in S} \underbrace{\Pr\{X^{(t+1)} = z \mid X^{(t)} = x, Y^{(t)} = y\}}_{\text{(b)}} \cdot \underbrace{\Pr\{Y^{(t)} = y\}}_{\text{(c)}} \tag{7.2}$$

where (a) is the probability of moving from the current state x to the next z, (b) the probability that z is offspring of x and y, and (c) the probability of sampling y from S [139].

Equation 7.2 defines the transition probabilities of a finite Markov chain representing a HC crossover random walk, but it is inconvenient because it leaves implicit the crossover and associated search space. Fortunately, in certain cases and under certain assumptions, ELT provides closed form expressions of Equation 7.2 in terms of the search space structure. The rest of this section focuses on the case that assumes terms (b) and (c) in Equation 7.2 follow uniform distributions [139]: all possible offspring are equally likely for given parents, and all individuals in the search space are equally likely to be sampled as second parent. A more general case [140] is briefly compared in Section 7.3.3 with other random walks seen in this chapter.

Definition for Any Recombination P-structure and Uniform Distribution

In ELT a recombination search space is a recombination P-structure $(X, \mathcal{R})$ denoted $\mathscr{R}$ (Definition 4.3) consisting of a finite set X of all candidate solutions and a structure imposed upon it by a certain interval function $\mathcal{R} : X \times X \to \mathcal{P}(X)$, which maps any two parents in X to their set of possible offspring in the power set $\mathcal{P}(X)$. That

is, $\mathcal{R}$ represents the support function of any crossover fulfilling the recombination P-structure axioms. Moreover, the associated recombination P-structure hypergraph $\text{hyp}_{\mathscr{R}}$ (Definition 4.6) has vertex set $V(\text{hyp}_{\mathscr{R}}) = X$, abbreviated V, and hyperedge set $E(\text{hyp}_{\mathscr{R}}) = \{\mathcal{R}(x, y) \mid x, y \in V,\ \mathcal{R}(x, y) \neq \emptyset\}$, abbreviated E (see Section 4.2).

A finite Markov chain is defined by a recombination P-structure $\mathscr{R}$ if uniform probability distributions are assumed in the right-hand side of Equation 7.2. The *set of states* of the Markov chain is the set of vertices $V = \{v_1, \ldots, v_n\}$ of the recombination P-structure hypergraph $\text{hyp}_{\mathscr{R}}$. The *transition probability matrix* $\mathbf{P}$ is defined entry-wise by *transition probabilities* $p_{x,z}$ [139] from the current state $x = v_i$ to the next state $z = v_j$ for $v_i, v_j \in V$ such that

$$\underbrace{p_{x,z}}_{\text{(a)}} \stackrel{\text{def}}{=} \sum_{y \in V} \underbrace{\frac{h_{z,(x,y)}}{|\mathcal{R}(x, y)|}}_{\text{(b)}} \cdot \underbrace{\frac{1}{|V|}}_{\text{(c)}} \ ; \tag{7.3}$$

where $h_{z,(x,y)}$ are the entries of the binary incidence matrix $\mathbf{H}$ (Equation 4.1) for the hypergraph $\text{hyp}_{\mathscr{R}}$ with $h_{z,(x,y)} = 1$ if and only if z is offspring of x, y. Notice Equation 7.3 is a specific case of Equation 7.2 with respect to terms (b) and (c): assuming that all offspring z are equally likely, term (b), and all individuals are equally likely as the second parent y, term (c).

Alternatively, $\mathbf{P}$ can be expressed in the equivalent and more concise form [139]

$$\mathbf{P} \stackrel{\text{def}}{=} \frac{1}{2\,|V|}\,\mathbf{S} \qquad \text{with entries } p_{x,z} \stackrel{\text{def}}{=} \frac{1}{2\,|V|}\,s_{x,z} \tag{7.4}$$

where $\mathbf{S}$ may be interpreted as a *generalised adjacency matrix* since, when its entries

$$s_{x,z} \stackrel{\text{def}}{=} 2 \sum_{y \in V} \frac{h_{z,(x,y)}}{|\mathcal{R}(x, y)|} \tag{7.5}$$

are non-zero, x and z are connected by a hyperedge $\mathcal{R}(x, y)$ in $\text{hyp}_{\mathscr{R}}$ (i.e. 'adjacent'). The factor $\frac{1}{2|V|}$ normalises $\mathbf{S}$ so the entries of $\mathbf{P}$ fall within $[0, 1]$.

Then, the Markov chain on a recombination P-structure $\mathscr{R}$ is determined by the pair $(\mathbf{p}^{(0)}, \mathbf{P})$. The row vector $\mathbf{p}^{(0)}$ is the initial probability distribution over the set of states, and the distribution after t time steps is given by the t-th power of $\mathbf{P}$: $\mathbf{p}^{(t)} = \mathbf{p}^{(0)}\,\mathbf{P}^t$. So $\mathbf{p}^{(0)}$ describes an initial solution, and $\mathbf{p}^{(t)}$ a solution after t steps, of a HC crossover random walk on a recombination P-structure $\mathscr{R}$.

Example: Uniform Recombination P-structure and Uniform Distribution

Recall Example 4.1 of the uniform recombination P-structure $\mathscr{R}_\Omega \stackrel{\text{def}}{=} (\mathcal{H}_q^n, \textsc{Uniform})$ specified by the function Uniform that represents the support function of the uniform crossover operator on n-dimensional q-ary Hamming sequences $\mathcal{H}_q^n$. Denote $\text{hyp}_{\mathscr{R}_\Omega}$ its hypergraph with vertex set $V = \mathcal{H}_q^n$ and hyperedge set $E = \{\textsc{Uniform}(x, y) \mid x, y \in V,\ \textsc{Uniform}(x, y) \neq \emptyset\}$.

For $\mathscr{R}_\Omega$ on binary sequences $(\mathcal{H}_2^n, \textsc{Uniform})$, the entries of the generalised adjacency matrix $\mathbf{S}$ (Equation 7.5) may be calculated more easily with the equivalent form $s_{x,z} = 2\left(\frac{3}{2}\right)^n 3^{-d_\text{H}(x,z)}$ (see Theorem 5 in [139]) where $d_\text{H}(x, z)$ is the Hamming distance between $x, z \in V$. Fixing $n = 3$ and $q = 2$ leads to the following Markov chain (see Appendix A.6 for a software implementation). States are vertices $V = \{v_1, \ldots v_8\}$ where $v_1 = \texttt{000}, \ldots, v_8 = \texttt{111}$. The transition probability matrix $\mathbf{P} = \frac{1}{2|V|}\,\mathbf{S} = \frac{1}{2\cdot 8}\,\mathbf{S}$ follows from Equation 7.4:

$$\mathbf{P} = \begin{bmatrix} \frac{27}{64} & \frac{9}{64} & \frac{9}{64} & \frac{3}{64} & \frac{9}{64} & \frac{3}{64} & \frac{3}{64} & \frac{1}{64} \\ \frac{9}{64} & \frac{27}{64} & \frac{3}{64} & \frac{9}{64} & \frac{3}{64} & \frac{9}{64} & \frac{1}{64} & \frac{3}{64} \\ \frac{9}{64} & \frac{3}{64} & \frac{27}{64} & \frac{9}{64} & \frac{3}{64} & \frac{1}{64} & \frac{9}{64} & \frac{3}{64} \\ \frac{3}{64} & \frac{9}{64} & \frac{9}{64} & \frac{27}{64} & \frac{1}{64} & \frac{3}{64} & \frac{3}{64} & \frac{9}{64} \\ \frac{9}{64} & \frac{3}{64} & \frac{3}{64} & \frac{1}{64} & \frac{27}{64} & \frac{9}{64} & \frac{9}{64} & \frac{3}{64} \\ \frac{3}{64} & \frac{9}{64} & \frac{1}{64} & \frac{3}{64} & \frac{9}{64} & \frac{27}{64} & \frac{3}{64} & \frac{9}{64} \\ \frac{3}{64} & \frac{1}{64} & \frac{9}{64} & \frac{3}{64} & \frac{9}{64} & \frac{3}{64} & \frac{27}{64} & \frac{9}{64} \\ \frac{1}{64} & \frac{3}{64} & \frac{3}{64} & \frac{9}{64} & \frac{3}{64} & \frac{9}{64} & \frac{9}{64} & \frac{27}{64} \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} \frac{27}{4} & \frac{9}{4} & \frac{9}{4} & \frac{3}{4} & \frac{9}{4} & \frac{3}{4} & \frac{3}{4} & \frac{1}{4} \\ \frac{9}{4} & \frac{27}{4} & \frac{3}{4} & \frac{9}{4} & \frac{3}{4} & \frac{9}{4} & \frac{1}{4} & \frac{3}{4} \\ \frac{9}{4} & \frac{3}{4} & \frac{27}{4} & \frac{9}{4} & \frac{3}{4} & \frac{1}{4} & \frac{9}{4} & \frac{3}{4} \\ \frac{3}{4} & \frac{9}{4} & \frac{9}{4} & \frac{27}{4} & \frac{1}{4} & \frac{3}{4} & \frac{3}{4} & \frac{9}{4} \\ \frac{9}{4} & \frac{3}{4} & \frac{3}{4} & \frac{1}{4} & \frac{27}{4} & \frac{9}{4} & \frac{9}{4} & \frac{3}{4} \\ \frac{3}{4} & \frac{9}{4} & \frac{1}{4} & \frac{3}{4} & \frac{9}{4} & \frac{27}{4} & \frac{3}{4} & \frac{9}{4} \\ \frac{3}{4} & \frac{1}{4} & \frac{9}{4} & \frac{3}{4} & \frac{9}{4} & \frac{3}{4} & \frac{27}{4} & \frac{9}{4} \\ \frac{1}{4} & \frac{3}{4} & \frac{3}{4} & \frac{9}{4} & \frac{3}{4} & \frac{9}{4} & \frac{9}{4} & \frac{27}{4} \end{bmatrix}. \tag{7.6}$$

Hence $\mathbf{P}$ alongside any initial distribution over the states determines a Markov chain on the uniform recombination P-structure $(\mathcal{H}_2^3, \textsc{Uniform})$. Notice the state diagram would be equivalent to a complete graph with a loop at each vertex. That is explained by the structure of the hypergraph $\text{hyp}_{\mathscr{R}_\Omega}$ in which any two vertices x and z are connected by at least one hyperedge (see Figure 4.3 for details). This reflects in all entries $s_{x,z}$ of the generalised adjacency matrix $\mathbf{S}$, and thus all entries of $\mathbf{P}$, being non-zero. Clearly the Markov chain properties fulfils all properties seen in Section 7.2.2:

(a) $\mathbf{P}$ is *non-negative* since all transition probabilities are greater than zero.

(b) $\mathbf{P}$ is *stochastic* since all rows add up to one.

(c) *Time-homogeneous* since the transition probabilities between states are con-

stant over time.

(d) *Reversible* since $\mathbf{P}$ is symmetric.

(e) *Irreducible* since any state can be reached from any other state after some time steps. Indeed, any state is reachable in one time step since each of the eight states has a transition to the other seven with non-zero probability.

Therefore, a HC crossover random walk described by the above Markov chain can generate in one step any individual (i.e. binary sequence) from any given 'father' at any given time. This is a consequence of properties (c)–(e) above and the assumptions on the uniform probability distributions over offspring and sampled 'mother' as second parent in Equation 7.2. For instance, take father $\mathtt{000}$, the offspring $\mathtt{011}$ can be generated only from recombinations $\textsc{Uniform}(\mathtt{000}, \mathtt{011})$ and $\textsc{Uniform}(\mathtt{000}, \mathtt{111})$, where $\mathtt{011}$ and $\mathtt{111}$ are the mothers. The probability $p_{\mathtt{000},\mathtt{011}}$ of such transition is $\frac{3}{64}$ from the transition probability matrix $\mathbf{P}$ in Equation 7.6. It is easy to see how $p_{\mathtt{000},\mathtt{011}}$ follows from such recombinations by expanding the entry $s_{\mathtt{000},\mathtt{011}}$ of the generalised adjacency matrix $\mathbf{S}$ with Equation 7.5:

$$s_{\mathtt{000},\mathtt{011}} = 2 \sum_{y \in V} \frac{h_{\mathtt{011},(\mathtt{000},y)}}{|\textsc{Uniform}(\mathtt{000}, y)|} \ .$$

Because $h_{\mathtt{011},(\mathtt{000},y)} = 0$ for all other mothers $y \in V \setminus \{\mathtt{011}, \mathtt{111}\}$, it follows

$$s_{\mathtt{000},\mathtt{011}} = 2 \left(\frac{h_{\mathtt{011},(\mathtt{000},\mathtt{011})}}{|\textsc{Uniform}(\mathtt{000}, \mathtt{011})|} + \frac{h_{\mathtt{011},(\mathtt{000},\mathtt{111})}}{|\textsc{Uniform}(\mathtt{000}, \mathtt{111})|} \right) = 2 \left(\frac{1}{4} + \frac{1}{8} \right) = \frac{3}{4} \ ,$$

so from Equation 7.4 the corresponding transition probability $p_{\mathtt{000},\mathtt{011}}$ in $\mathbf{P}$ is

$$p_{\mathtt{000},\mathtt{011}} = \frac{1}{2\,|V|}\, s_{\mathtt{000},\mathtt{011}} = \frac{1}{2 \cdot 8} \frac{3}{4} = \frac{3}{64} \ .$$

7.3.3 Finite Markov Chain: Generalisation

The finite Markov chain in Section 7.3.2 is generalised by [140] to model HC crossover random walks where no uniform probability distributions have to be assumed in Equation 7.2 over offspring or their parents. To achieve it, other assumptions are introduced and the generality of other aspects is sacrificed:

- It assumes recombination P-structures based on strings, such as the uniform and the one-point recombination P-structures (Examples 4.1–4.2), rather than general recombination P-structures (Definition 4.3).

- It assumes Geiringer's *linkage equilibrium* [50] distribution in Equation 7.2 from which the second parent is sampled for HC crossover (Figure 7.3). This distribution does not have to be uniform and may be defined over a subset rather than the entire set of possible individuals as Section 7.3.2 does. But it requires there is no genetic interdependence (i.e. epistasis) between the alleles of the individuals' genotypes: for each string (i.e. individual), the values at each position are statistically independent from the values at other positions. So if X is a random variable that represents some individual $x = (x_1, \ldots, x_n)$ sampled from a population under linkage equilibrium, and $f_i(x_i)$ is the frequency of x_i at position i for such population, then $\Pr\{X = x\} \stackrel{\text{def}}{=} \prod_i f_i(x_i)$.

- The finite Markov chain is reversible (see Lemma 2 in [140]), though not time-homogeneous nor irreducible necessarily. That is, transition probabilities are symmetric, but they may change over time and become zero for some states (i.e. states may become unreachable).

In summary, ELT considers two major types of random walks (Table 7.1). For mutation search spaces, random walks on graphs (Section 7.2). For recombination search spaces (i.e. recombination P-structures), HC crossover random walks assuming either a uniform (Section 7.3.2) or linkage equilibrium probability distribution (Section 7.3.3).

Markov chain properties	Random walk based on mutation	HC crossover random walk (1)	HC crossover random walk (2)
Time-homogeneous	●	●	○
Reversible†	●	●	●
Irreducible	●	●	○

(1) Uniform distribution (2) Linkage equilibrium distribution

(●) Property holds (○) Property may not hold

† Assuming symmetric neighbourhoods and recombination P-structures on strings

Table 7.1. Types of (crossover) random walks and key properties fulfilled by their respective finite Markov chains.

Chapter 8

A Qualitative Framework for Abstract Interval Convex Search of Evolutionary Algorithms

This chapter is an original major contribution of this thesis, extending a paper that I co-authored with Moraglio [49]. This chapter presents a qualitative framework built upon the crossover classification in Chapter 5, to systematically study the abstract interval convex evolutionary search of a general EA class that unifies geometric-crossover EAs (reviewed in Chapter 6) and HC crossover random walks for recombination P-structures (reviewed in Chapter 7).

8.1 Introduction

The classification of crossovers proposed in Chapter 5 is a formal system to understand crossovers across problems and representations with respect to the axioms of geometric crossovers and recombination P-structures. Building upon this crossover classification, this chapter now develops a framework for an evolutionary search model generalising those of the GF and ELT reviewed in Chapters 6 and 7.

In the GF, formal metric EAs (Section 6.1) are a general class of EAs defined with geometric mutation and crossover operators independent of a specific problem and representation. Among them, geometric-crossover EAs (without mutation) stand out: their behaviour is always abstract convex evolutionary search (Section 6.3), and their runtime is provably polynomial on problems with certain abstract convex landscapes according to Moraglio and Sudholt [104]. However, EAs using non-geometric crossovers or mutation do not do abstract convex evolutionary search [101].

By contrast, ELT models recombination-like search through headless-chicken (HC) crossover random walks (Section 7.3). A HC crossover (or macro-mutation) can be any recombination P-structure where one of the parents is taken from a previous recombination and the other is randomly sampled from a prescribed probability distribution (possibly over the entire solution set). No populations, selection nor replacement mechanisms as in GAs are actually used though: formalising population-based EAs for recombination P-structures is an open challenge in ELT [85].

Neither geometric-crossover EAs nor recombination P-structure random walks are fully satisfactory as a general and realistic model of EAs. The former lack non-geometric crossovers, some of which are useful like Koza's subtree swap [86] or Davis's order [31], and the ability to mutate individuals, which may prevent premature convergence. The latter are not EAs really. This chapter aims towards a general EA class that integrates both approaches, namely geometric-crossover EAs and recombination P-structure random walks, and whose abstract behaviour across problems and representations remains analogous to abstract convex evolutionary search. The next fundamental open question frames it concisely.

Question 8.1. *Is there any class of EAs, other than geometric-crossover EAs, based on recombination P-structures and whose behaviour is analogous to abstract convex evolutionary search (Proposition 6.1)? Is it still true, if such EAs use mutation?*

This chapter proves that abstract convex evolutionary search extends to a class called formal interval EAs. Formal interval EAs can use a wide range of crossovers including geometric ones, non-geometric like some recombination P-structures, and certain forms of crossover with mutation. Furthermore, recombination P-structure EAs, a subclass of formal interval EAs, provides ELT with the means to model population-based EA using recombination P-structures for the first time.

The brief answer to Question 8.1 will be that the abstract behaviour of formal interval and recombination P-structure EAs is theoretically analogous though not always equivalent to geometric-crossover EAs. By comparing the abstract convexities [146] intrinsic to geometric and non-geometric crossovers, this chapter will show that some non-geometric crossovers have 'degenerate' abstract convexities leading to degenerate types of abstract convex search which no longer describe the actual behaviour of populations. Specifically, the contributions of this chapter are:

1. Extending the classification of crossovers in Chapter 5 with interval operators of finite interval spaces. This extension also explains why Stadler and others [135] observed that Koza's subtree swap crossover [86] is similar to unequal one-point crossover [131] in that neither lie in a metric space. (Section 8.2.1.)

2. Defining formal interval EAs and their subclass recombination P-structure EAs based on the extended classification, which generalise geometric-crossover EAs using geometric recombination P-structures. (Section 8.2.2.)

3. Explaining how various fundamental notions on abstract interval convexity relate to crossovers in the extended classification. (Section 8.3.)

4. Proving that any formal interval-crossover EA (including any recombination P-structure EA) does a generalised form of abstract convex evolutionary search called abstract interval convex evolutionary search. (Section 8.4.)

5. Illustrating how crossovers in the extended crossover classification affect convergence of abstract interval convex evolutionary search. Degenerate and non-degenerate cases are discriminated via three coarse-grained types of abstract interval convex evolutionary search. (Section 8.4.1.)

6. Proving abstract interval convex evolutionary search forms a sequence of nested generalised schemas or invariant subsets in Mitavskiy's sense [98, 99]. (Section 8.4.2)

Section 8.5 discusses this chapter's contributions. Section 8.6 concludes this chapter.

8.2 Towards Formal Interval EAs

Before defining formal interval EAs, we need to introduce the class of generalised crossover operators characteristic of formal interval EAs called *finite interval operators* (Definition 8.1) within the universal class $\mathcal{U}$ of all possible crossovers. We focus on their support structure, describing what offspring are reachable from parents rather than how probable, similar to the formalisation of geometric crossovers and recombination P-structures in Chapters 3 and 4 respectively.

Definition 8.1 (**Finite interval space** [146])**.** Let $I : X \times X \to \mathcal{P}(X)$ be any function defined on any finite set X such that $\forall x, y \in X$: (I) extensitivity: $x, y \in I(x, y)$; and, (II) symmetry: $I(x, y) = I(y, x)$. Then, I is a *finite interval operator* on X, $I(x, y)$ is the *(finite) interval between* x and y, and (X, I) is a *finite interval space*. The *class* of all finite interval operators I is denoted $\mathcal{I}_{fin}$ and its universal complement $\overline{\mathcal{I}_{fin}} \stackrel{\text{def}}{=} \mathcal{U} \setminus \mathcal{I}_{fin}$.

Finite interval operators will allow us to address Question 8.1 for three main reasons:

1. $\mathcal{I}_{fin}$ is a superclass of recombination P-structures $\mathcal{RP}$, and $\mathcal{RP}$ includes certain non-geometric crossovers as well as certain geometric crossovers, namely geometric recombination P-structures (see Chapter 5).

2. $\mathcal{I}_{fin}$ includes certain forms of crossover that mutate individuals and are not covered by recombination P-structures.

3. All finite interval operators have a corresponding abstract interval convexity.

The next Section 8.2.1 extends the crossover classification presented in Chapter 5 to justify the first two reasons; the third reason is covered later in Section 8.3.

8.2.1 A Extended Classification of Crossovers

This section presents a extended classification of crossovers (Figure 8.1) by proving through Theorems 8.1–8.6 the main class relations that incorporate the class of finite interval operators $\mathcal{I}_{fin}$ and two additional crossovers: the unequal one-point crossover by Shpak and Wagner [131, 135] and ball-mutation segment crossover defined next.

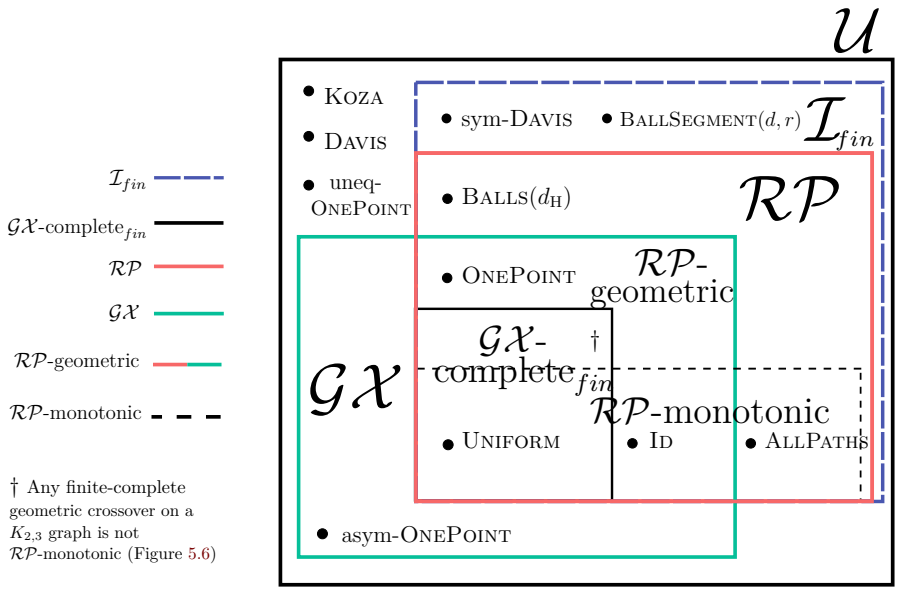


Figure 8.1. Extended classification within the universal class $\mathcal{U}$ of crossovers. *Classes*: geometric crossovers $\mathcal{GX}$, recombination P-structures $\mathcal{RP}$ and finite interval operators $\mathcal{I}_{fin}$. *Subclasses*: finite-complete geometric crossovers ($\mathcal{GX}$-complete$_{fin}$), geometric recombination P-structures ($\mathcal{RP}$-geometric) and monotonic recombination P-structures ($\mathcal{RP}$-monotonic). *Crossover examples*: unbiased uniform (Uniform), one-point (OnePoint), asymmetric one-point (asym-OnePoint), unequal one-point (uneq-OnePoint), Koza's subtree swap (Koza), Davis's order (Davis), symmetric Davis's order (sym-Davis), all-Hamming-paths (AllPaths($H(n, q)$)), intersecting-Hamming-balls (Balls(d_H)), ball-mutation segment (BallSegment(d, r)), and identity (Id).

Definition 8.2 (**Ball-mutation segment crossover**)**.** Let (X, d) be any metric space and arbitrary parents $x, y \in X$. Then, the *ball-mutation segment crossover* is defined as $\textsc{BallSegment}(d, r) : X \times X \to \mathcal{P}(X)$, $(x, y) \mapsto \cup\{\bar{B}_d(z, r) \mid z \in [x, y]_d\}$ parametrised with metric d and radius $r \in \mathbb{R}_{>0}$.

A ball-mutation segment crossover, illustrated in Figure 8.2, applies a d-metric ball mutation to all possible offspring in the d-metric segment $[x, y]_d$ between two parents x, y, that is the offspring of a complete geometric crossover (Definition 3.3). Consider for example, on a binary Hamming metric space, the *Hamming ball-mutation segment crossover* $\textsc{BallSegment}(d_{\mathrm{H}}, 1)$ parametrised with Hamming distance and ball radius one; see Appendix A.4 for a software implementation. To apply $\textsc{BallSegment}(d_{\mathrm{H}}, 1)(x, y)$ is equivalent to recombine parents x, y under uniform crossover and then return all neighbours under single-bit flip mutation. So $\{\texttt{000}, \texttt{100}\}$ are the offspring of parents $\texttt{000}$ and $\texttt{100}$ under uniform crossover, and those offspring along with their neighbours under single-bit flip mutation are $\{\texttt{000}, \texttt{001}, \texttt{010}, \texttt{100}\} \cup \{\texttt{100}, \texttt{000}, \texttt{101}, \texttt{110}\}$ (Figure 8.2b).

$\textsc{BallSegment}(d_2, r)(x, y)$ **(a)** $\textsc{BallSegment}(d_{\mathrm{H}}, 1)(\texttt{000}, \texttt{100})$ **(b)**

Figure 8.2. Ball-mutation segment crossover in two-dimensional Euclidean metric space $(\mathbb{R}^2, d_2)$ in Figure 8.2a ('shaded region'). Ball-mutation segment crossover in three-dimensional binary Hamming space $(\mathcal{H}_2^3, d_{\mathrm{H}})$ in Figure 8.2b ('black lines').

From the example above, it is not difficult to see why $\textsc{BallSegment}(d, r)$ belongs to $\mathcal{I}_{fin}$ but not to $\mathcal{RP}$ nor $\mathcal{GX}$ (Theorem 8.1). Note that recombination P-structures $(X, \mathcal{R})$ require the fix-point axiom $\forall x \in X : \mathcal{R}(x, x) = \{x\}$, known as the purity inbreeding property for geometric crossovers (Proposition 3.2). Clearly, even if a crossover fulfils them, applying a non-degenerate[1] mutation to the offspring inevitably breaks the fix-point axiom as well as the purity inbreeding property (Lemma 8.1). Finite interval operators do not require such fix-point axiom or purity inbreeding property.

[1]The input and output individuals of non-degenerate mutation operators are always distinct.

Lemma 8.1. Let $\xi : X \times X \to \mathcal{P}(X)$ and $\mu : X \to \mathcal{P}(X)$ be support functions of an arbitrary crossover and a mutation operator, respectively, defined on a non-empty finite set X such that $\forall x \in X$: $\xi(x, x) = \{x\}$ and $\mu(x) \neq \{x\}$. Define an operator $\textsc{Op}(x, y) \stackrel{\text{def}}{=} \{\mu(z) \mid z \in \xi(x, y)\}$ for all parents $x, y \in X$. Then, Op is not a geometric crossover and $(X, \textsc{Op})$ is not a recombination P-structure.

Proof. Op fails the fix-point axiom of recombination P-structures (Definition 4.3): $\textsc{Op}(x, x) = \{\mu(z) \mid z \in \xi(x, x)\} = \{\mu(z) \mid z \in \{x\}\} = \{\mu(x)\} \neq \{x\}$. Hence Op also fails the purity inbreeding property of geometric crossovers (Proposition 3.2). □

Theorem 8.1 ($\textsc{BallSegment}(d, r) \in \overline{\mathcal{GX}}$, $\overline{\mathcal{RP}}$, $\mathcal{I}_{fin}$)**.** Let the ball-mutation segment crossover $\textsc{BallSegment}(d, r)$, with any radius $r \in \mathbb{R}_{>0}$, on a finite and connected metric space (X, d) where $|X| \geq 2$. Then, $\textsc{BallSegment}(d, r)$ is not a geometric crossover, and $(X, \textsc{BallSegment}(d, r))$ is not a recombination P-structure but a finite interval space.

Proof.

1. By Lemma 8.1 and Definition 8.2, $\textsc{BallSegment}(d, r)$ is not a geometric crossover and $(X, \textsc{BallSegment}(d, r))$ is not a recombination P-structure. Notice for any non-zero radius and any $x \in X$: $\textsc{BallSegment}(d, r)(x, x) = \cup\{\bar{B}_d(z, r) \mid z \in [x, x]_d\} = \cup\{\bar{B}_d(z, r) \mid z \in \{x\}\} = \{\bar{B}_d(x, r)\} \neq \{x\}$. The last inequality is true by the assumption of (X, d) being connected and containing at least two points (i.e. every closed metric ball with non-zero radius contains at least one point other than its centre).

2. $(X, \textsc{BallSegment}(d, r))$ is a finite interval space because X is a finite set and $\textsc{BallSegment}(d, r)$ fulfils the extensivity and symmetry axioms of interval spaces (Definition 4.2). Extensivity follows from Definition 3.2 since metric segments include their extremes (i.e. $x, y \in [x, y]_d$) and closed metric balls their centres (i.e. $z \in \bar{B}_d(z, r)$), so $x, y \in \cup\{\bar{B}_d(z, r) \mid z \in [x, y]_d\} = \textsc{BallSegment}(d, r)(x, y)$. Symmetry follows from the symmetry of metric segments: $\textsc{BallSegment}(d, r)(x, y) = \cup\{\bar{B}_d(z, r) \mid z \in [x, y]_d\} = \cup\{\bar{B}_d(z, r) \mid z \in [y, x]_d\} = \textsc{BallSegment}(d, r)(y, x)$. □

Despite being so general that admits crossovers with mutation (e.g. ball-mutation segment crossover), the class $\mathcal{I}_{fin}$ of finite interval operators does not exhaust all crossovers. One of them is the *unequal one-point crossover* (Definition 8.3) proposed by Shpak and Wagner [131] to model gene duplication in evolutionary biology. Next, we prove that unequal one-point crossover is non-geometric and falls outside the class $\mathcal{I}_{fin}$ alongside Koza's subtree swap crossover. This justifies why Koza's crossover is

similar to unequal one-point crossover in that both are undefined on metric spaces or non-metrisable as Stadler and others [135] observed.

Definition 8.3 (**Unequal one-point crossover** [131, 135])**.** Let X be any non-empty set and $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_m)$ be sequences in X of arbitrary lengths $n, m \in \mathbb{N}_0$, where a zero-length sequence is the *empty sequence* ε. Denote the *prefix* $x_{[i]} \stackrel{\text{def}}{=} (x_1, x_2, \ldots, x_i)$ and *suffix* $x^{[j]} \stackrel{\text{def}}{=} (x_{j+1}, x_{j+2}, \ldots, x_n)$ subsequences of x, where $x_{[0]} = x^{[n]} = \varepsilon$, $x_{[n]} = x^{[0]} = x$; also, denote $xy \stackrel{\text{def}}{=} (x_1, \ldots, x_n, y_1, \ldots, y_m)$ the *concatenation* of x and y. The *unequal one-point crossover* produces all concatenations up to length $n+m$ of prefixes and suffixes of parents $x, y \in X$: uneq-OnePoint(x, y) $\stackrel{\text{def}}{=} \{x_{[i]}y^{[j]} \mid 0 \leq i \leq n, 0 < j \leq m\} \cup \{y_{[j]}x^{[i]} \mid 0 \leq i \leq n, 0 < j \leq m\}$.

Length	uneq-OnePoint$(\texttt{01}, \varepsilon)$	uneq-OnePoint$(\texttt{01}, \texttt{1})$	uneq-OnePoint$(\texttt{00}, \texttt{11})$
0	ε	ε	ε
1	0 1	0 1	0 1
2	01	01 10	00 01 10 11
3		011 101	001 011 100 110
4			0011 1100

Table 8.1. Offspring binary Hamming sequences, ordered by their length, obtained under unequal one-point crossover for: uneq-OnePoint$(\texttt{01}, \varepsilon)$, uneq-OnePoint$(\texttt{01}, \texttt{1})$ and uneq-OnePoint$(\texttt{00}, \texttt{11})$, where ε is the empty sequence.

Unlike the traditional one-point crossover (Definition 4.5), the *unequal one-point crossover* can recombine and produce sequences of variable length, including the zero-length empty sequence, as exemplified in Table 8.1. This makes unequal crossover similar to Koza's subtree swap (Definition 5.1) in that both can increase the dimensionality of the space and produce offspring distinct from two identical parents; compare Table 8.1 and Figure 8.3 below. That is why neither of them induce a finite interval space in general (Theorems 8.2 and 8.3) nor fulfil the purity inbreeding property of geometric crossovers (Theorems 5.6 and 8.2).

Theorem 8.2 (uneq-OnePoint $\in \overline{\mathcal{GX}}$, $\overline{\mathcal{I}_{fin}}$)**.** Let X be any non-empty set. Then, the unequal one-point crossover uneq-OnePoint is not a geometric crossover, and $(X,$ uneq-OnePoint$)$ is not a finite interval space.

Proof. 1. By Definition 8.3, uneq-OnePoint can produce offspring different from parents even if these are identical: uneq-OnePoint$(x, x) \neq \{x\}\ \forall x \in X$ in general [131]. Hence uneq-OnePoint is non-geometric for violating the purity inbreeding property of geometric crossovers (Proposition 3.2).

2. In general, $(X,$ uneq-OnePoint$)$ is not a finite interval space due to its Definition 8.3. Note uneq-OnePoint can produce sequences of arbitrary variable length (see Table 8.1), so even if a finite X is assumed, uneq-OnePoint could produce offspring outside X [131]. □

Theorem 8.3 (Koza $\in \overline{\mathcal{I}_{fin}}$)**.** Let T be any non-empty set of labelled ordered trees. Then, $(T,$ Koza$)$ is not a finite interval space.

Proof. In general, $(T,$ Koza$)$ is not a finite interval space by Definition 5.1. Koza produces all possible subtree swaps for any two parent trees $x, y \in T$ with vertex sets $V(x), V(y)$: Koza$(x, y) = \bigcup_{\substack{i \in V(x),\\ j \in V(y)}}$ Koza$(i, j)(x, y)$. With this definition, Koza can produce offspring trees outside T, if assumed finite, due to their larger dimensionality. For example, let $T = \{t_1, t_2\}$ as in Figure 8.3a, then one offspring pair produced by Koza(t_1, t_2) is Koza$(2, 1)(t_1, t_2)$ in Figure 8.3b, where the none of them are in T and the first offspring is clearly larger than both parents. □

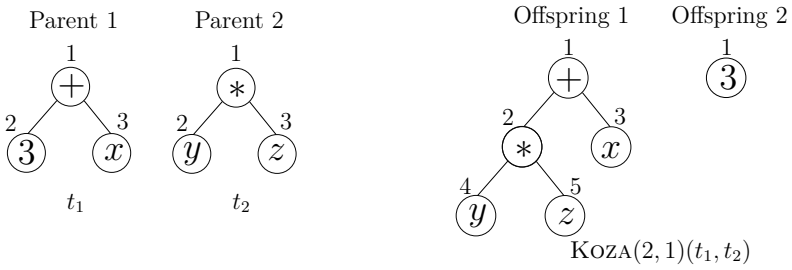


Figure 8.3. Koza's subtree swap can produce offspring (Figure 8.3b) of dimensionality larger than parents (Figure 8.3a), so it cannot induce a finite interval space in general.

Like uneq-OnePoint and Koza, Davis order crossover (Definition 5.2) is neither a finite interval operator though for another reason: it is asymmetric. According to Theorem 8.4, Davis belongs to $\overline{\mathcal{I}_{fin}}$ but its symmetric version sym-Davis (Definition 5.3) belongs to $\mathcal{I}_{fin}$. An analogous situation occurs between the asymmetric one-point crossover (Definition 3.6) and the (symmetric) one-point crossover (Definition 4.5) due to Theorem 8.5. This highlights how crucial symmetry (or lack thereof) may be to distinguish seemingly identical versions of the same crossover.

Theorem 8.4 (DAVIS $\in \overline{\mathcal{I}_{fin}}$, sym-DAVIS $\in \mathcal{I}_{fin}$)**.** Let $\mathcal{S}_X$ be permutations of any non-empty finite set X. Then, $(\mathcal{S}_X, \text{DAVIS})$ is not a finite interval space; $(\mathcal{S}_X, \text{sym-DAVIS})$ is a finite interval space.

Proof. $(\mathcal{S}_X, \text{DAVIS})$ is not an interval space for the same reason Theorem 5.7 proved it not to be a recombination P-structure: it fails to be symmetric. $(\mathcal{S}_X, \text{sym-DAVIS})$ is trivially an interval space by its Definition 5.3: it is symmetric, and since it produces all possible offspring across all possible crossover sections (including the entire sequence) both parents are generated as offspring (i.e. extensivity holds). □

Theorem 8.5 (asym-ONEPOINT $\in \overline{\mathcal{I}_{fin}}$)**.** Let $\mathcal{H}_q^n$ be n-dimensional q-ary Hamming sequences ($q \geq 2$) and asym-ONEPOINT the asymmetric one-point crossover. Then, $(\mathcal{H}_q^n, \text{asym-ONEPOINT})$ is not a finite interval space.

Proof. Trivial, for the same reason Theorem 5.5 proved asym-ONEPOINT not to be recombination P-structure: it does not fulfil the symmetry axiom. □

Theorem 8.6 (**Crossover classification extension**)**.** Consider the original classification of crossovers within the universal class $\mathcal{U}$, the class $\mathcal{I}_{fin}$ of finite interval operators and its complement $\overline{\mathcal{I}_{fin}} = \mathcal{U} \setminus \mathcal{I}_{fin}$. The following relationships hold:

(a) $\overline{\mathcal{I}_{fin}} \neq \emptyset$. *The class of crossovers that are not finite interval operators is not empty.*

(b) $\mathcal{RP} \subset \mathcal{I}_{fin}$. *The class of crossovers that are recombination P-structures is a strict subclass of those crossovers that are finite interval operators.*

(c) $\mathcal{GX} \neq \mathcal{I}_{fin}$. *Geometric crossovers and finite interval operators represent distinct classes of crossovers.*

(d) $\overline{\mathcal{GX} \cup \mathcal{I}_{fin}} \neq \emptyset$. *The class of crossovers that are neither geometric crossovers nor finite interval operators is not empty.*

(e) $\mathcal{GX} \cap \mathcal{I}_{fin} \neq \emptyset$. *The class of crossovers that are both geometric crossovers and finite interval operators is not empty.*

Proof. (a) Unequal one-point, Koza's subtree swap, Davis's order and asymmetric one-point crossovers all belong to $\overline{\mathcal{I}_{fin}}$ (Theorems 8.2–8.5).

(b) By Definitions 4.2–4.3, any recombination P-structure operator is a finite interval operator, so $\mathcal{RP} \subseteq \mathcal{I}_{fin}$. By Theorem 5.7 and Theorem 8.4 if follows $\mathcal{RP} \not\ni$ sym-DAVIS $\in \mathcal{I}_{fin}$, so $\mathcal{RP} \subset \mathcal{I}_{fin}$.

(c) asym-OnePoint crossover belongs to the class of geometric crossovers $\mathcal{GX}$ (Theorem 5.5) but not to the class of finite interval operators $\mathcal{I}_{fin}$ due to asymmetry (Theorem 8.5). Conversely, sym-Davis crossover belongs to $\mathcal{I}_{fin}$ (Theorem 8.4) but not to $\mathcal{GX}$ (see Remark 5.1 and Theorem 5.15).

(d) By Theorems 5.6–5.7 and Theorems 8.2–8.4, unequal one-point, Koza's subtree swap and Davis's order crossovers are not geometric crossovers nor finite interval operators.

(e) $\mathcal{GX} \cap \mathcal{RP} \neq \emptyset$ holds because of Theorem 5.15, and $\mathcal{RP} \subset \mathcal{I}_{fin}$ holds because of (b). Therefore, $\mathcal{GX} \cap \mathcal{I}_{fin} \neq \emptyset$. □

8.2.2 Defining Formal Interval EAs

For each crossover (class) in the extended crossover classification (Section 8.2.1), one can think of specific EAs defined with those crossovers and, by abstraction, treat them as a class. For example, geometric-crossover EAs (Definition 6.3) comprise all specific EAs using any geometric crossover in the class $\mathcal{GX}$ and no mutation.

By the extended crossover classification, geometric crossovers in the non-empty class $\mathcal{GX} \cap \mathcal{RP} = \mathcal{RP}$-geometric are a subclass of recombination P-structures $\mathcal{RP}$ and of finite interval operators $\mathcal{I}_{fin}$ in turn. This hierarchical relationship between $\mathcal{GX}$, $\mathcal{RP}$ and $\mathcal{I}_{fin}$ is the fundamental reason why Definitions 8.4–8.5 of *formal interval* EAs and *recombination P-structure* EAs are proper generalisations over geometric-crossover EAs based on $\mathcal{RP}$-geometric, called geometric recombination P-structure EAs. They should be treated as formal mathematical objects relying only on the axioms of finite interval spaces and recombination P-structures. Table 8.2 summarises the correspondence between such crossover classes and EA classes.

EA class	Crossover class	Search space structure
Formal interval EA	$\mathcal{I}_{fin}$	Finite interval space
Recombination P-structure EA	$\mathcal{RP}$	Recombination P-structure
Geometric recombination P-structure EA	$\mathcal{GX} \cap \mathcal{RP}$	Finite graphic metric space

Table 8.2. Key EA classes associated with crossover classes in the extended crossover classification and their corresponding search space structures.

Definition 8.4 (Formal interval EA). A *formal interval evolutionary algorithm* is any EA using a crossover whose support function is given by the interval operator I of a fixed but unspecified finite interval space (X, I), where the finite set X describes all candidate solutions and I imposes a structure on X.

Definition 8.5 (**Recombination P-structure EA**)**.** A *recombination P-structure evolutionary algorithm* is any formal interval EA (without mutation) restricted from finite interval spaces to a fixed but unspecified recombination P-structure $(X, \mathcal{R})$.

The key components of formal interval EAs and recombination P-structure EAs are defined similarly to those of geometric-crossover EAs seen in Section 6.1:

- *Populations* are multi-sets, the *fitness function* is fixed but unspecified, *selection* and *replacement* population operators as for geometric-crossover EAs.
- *Recombination/Crossover*: accepts an input population and outputs another population by applying any crossover either in $\mathcal{RP}$ or $\mathcal{I}_{fin}$ any desired number of times to individuals in the input population. For recombination P-structure EAs, the underlying crossover belongs to $\mathcal{RP}$ and mutation cannot occur since it violates the fix-point axiom of recombination P-structures (see Lemma 8.1). For formal interval EAs, the crossover belongs to $\mathcal{I}_{fin}$ and mutation may occur (e.g. ball-mutation segment crossover in Definition 8.2).
- No *mutation* population operator is explicitly used, but in formal interval EAs mutation may occur during recombination.

Specific implementations of those components determine a specific instance of a formal interval EA, or recombination P-structure EA, with a corresponding and possibly distinct specific behaviour. Nevertheless, all those instances share a common behaviour called *abstract interval convex evolutionary search*. Section 8.4 will show it generalises abstract (geodesically) convex evolutionary search of geometric-crossover EAs (Proposition 6.1). First, Section 8.3 needs to introduce interval convexities, more general than geodesic ones, associated with crossovers in $\mathcal{RP}$ or $\mathcal{I}_{fin}$.

Definition 8.6 (**Abstract interval convex evolutionary search**)**.** *Abstract interval convex evolutionary search* is the abstract behaviour of a formal interval EA across all finite interval spaces (X, I). That is, the behaviour common to all specific EAs obtained from the formal interval EA by specifying set of all candidate solutions X and an interval operator I.

8.3 Interval Convexities of Crossovers

Geometric crossovers are naturally associated with geodesic convexities of metric spaces (Section 6.2). From the classification of crossovers, whether the original (Section 5.2) or its extension (Section 8.2.1), it is clear that geometric recombination

P-structures are associated with geodesic convexities as well since they are a specific case of geometric crossovers by Definition 4.8. But what is a natural convexity for other recombination P-structures $\mathcal{RP}$ or the superclass of finite interval operators $\mathcal{I}_{fin}$? The answer is *interval convexities* (Definition 8.7) because:

- $\mathcal{RP}$ is a subclass of $\mathcal{I}_{fin}$, hence any crossover in either of them is a finite interval operator of some finite interval space.
- Every (finite) interval space has a corresponding interval convexity [146].

Together, these two facts imply that every crossover in $\mathcal{RP}$ and $\mathcal{I}_{fin}$ is associated with an interval convexity. It will be called either *finite interval convexity*, if it assumes a finite interval space, or *recombination P-structure convexity*, if it assumes a recombination P-structure.

Definition 8.7 (**Interval convex space** [146])**.** Let (X, I) be any interval space and $x, y, z \in X$ arbitrary points. Then, z is in the *interval between* x and y, if and only if $z \in I(x, y)$. A subset $C \subseteq X$ is *interval convex*, if and only if all intervals between pairs of points in C are also in C; that is, $\forall x, y \in C : I(x, y) \subseteq C$. The family $\mathcal{C}$ of all interval convex sets C in X is the *interval convexity* of (X, I), and the pair $(X, \mathcal{C})$ is an *interval convex space*.

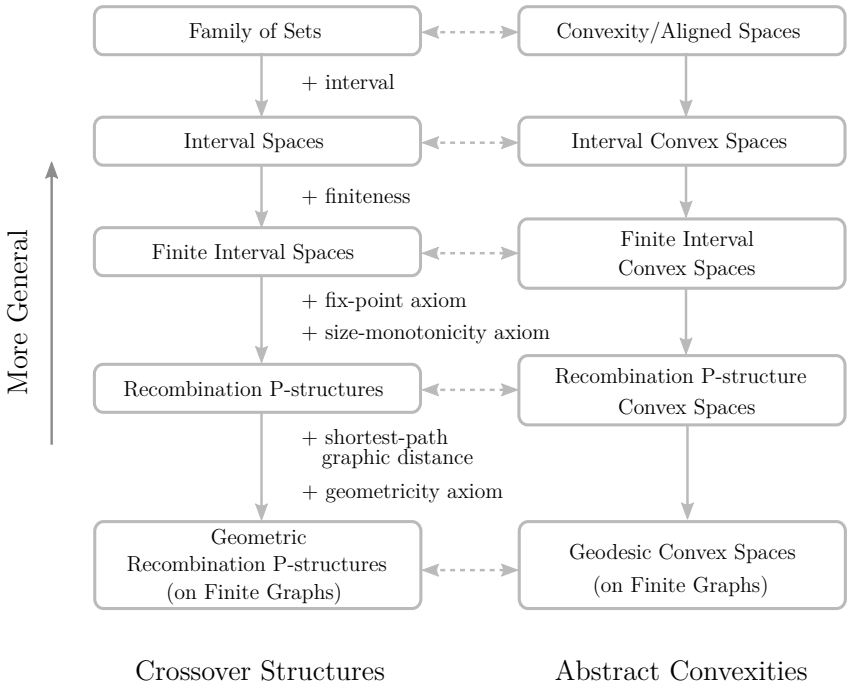


Figure 8.4. Ladder of hierarchical relationships between interval spaces and associated abstract convexities according to their level of generality. A 'downward arrow' between any two levels indicates that the level below inherits all of the attributes of the one above and incorporates the new attributes '+'.

Depending on what assumptions are made, interval convexities may be specialised or generalised to other convexities (Figure 8.4). For instance, finite interval convexities specialise to recombination P-structure convexities by assuming the fix-point and size-monotonicity axioms of recombination P-structures. Moreover, recall from Theorem 5.1 that all geodesic intervals on finite metric spaces are recombination P-structures, so recombination P-structure convexities can be further specialised to geodesic convexities (Definition 6.4) on finite graphs by assuming a graphic shortest-path distance. By contrast, dropping the notion of interval in interval spaces leaves us with just a family of sets, leading to *aligned spaces* (Definition 8.8) as the most abstract convexities. Although other abstract spaces exist beyond aligned spaces (e.g. pre-topologies) [134, 135], interval convex spaces and the Kuratowski's axioms inherited from aligned spaces already suffice to prove abstract interval convex search of formal interval EAs later in Section 8.4.

Definition 8.8 (**Convexity/Aligned space** [146])**.** A family $\mathcal{C}$ of subsets of a set X is called a *convexity* or *alignment* on X if: (I) the empty set $\emptyset$ and the universal set X are in $\mathcal{C}$; (II) $\mathcal{C}$ is stable under arbitrary intersections, that is if $\mathcal{D} \subseteq \mathcal{C}$, then $\bigcap \mathcal{D} \in \mathcal{C}$; and, (III) $\mathcal{C}$ is stable under arbitrary nested unions, that is if $\mathcal{D} \subseteq \mathcal{C}$ is totally ordered by inclusion, then $\bigcup \mathcal{D} \in \mathcal{C}$. The pair $(X, \mathcal{C})$ is called *convex structure*, *convexity space* or *aligned space*, and the members of $\mathcal{C}$ are *convex sets*.

Definition 8.9 (**Convex hull: Kuratowski's axioms** [94, 146])**.** Let $(X, \mathcal{C})$ be an aligned space and arbitrary subsets $A, B \in \mathcal{P}(X)$. The convex hull closure $co : \mathcal{P}(X) \to \mathcal{P}(X)$, $co(A) \stackrel{\text{def}}{=} \bigcap\{C \mid A \subseteq C \in \mathcal{C}\}$, fulfils the following *Kuratowski axioms*: (I) normalisation: $co(\emptyset) = \emptyset$; (II) extensivity: $A \subseteq co(A)$; (III) isotony: $A \subseteq B \implies co(A) \subseteq co(B)$; and, (IV) idempotency: $co(co(A)) = co(A)$.

For any of the abstract convexities in Figure 8.4, a convex hull closure of a set can be defined as the intersection of all convex supersets of that set; all of them conforming to Kuratowski's axioms (Definition 8.9). Convex hulls in interval spaces (for short, *interval convex hulls*) have another equivalent definition that makes explicit how a crossover's support function induces an interval convexity as follows.

Definition 8.10 (**[Recursive] pre-hull operator** [112, 146])**.** Let (X, I) be any interval space and $S \in \mathcal{P}(X)$ any subset in the power set of X. A *(recursive) pre-hull operator* for (X, I) and some natural number $k \in \mathbb{N}_0$ is defined on S by the

recursive function $cl_I^k : \mathcal{P}(X) \to \mathcal{P}(X)$

$$cl_I^0(S) \stackrel{\text{def}}{=} S \ ,$$
$$cl_I^k(S) \stackrel{\text{def}}{=} \cup \{I(x,y) \mid x,y \in cl_I^{k-1}(S)\} \ .$$

The smallest k such that $cl_I^{k+1}(S) = cl_I^k(S)$ is the *closure iteration number*[2] of S denoted $cin_I(S) \stackrel{\text{def}}{=} \min_{k\in\mathbb{N}_0} cl_I^{k+1}(S) = cl_I^k(S)$. The *maximum closure iteration number* of X is $cin_I^*(X) \stackrel{\text{def}}{=} \max_{S\subseteq X} cin_I(S)$. Note $cin_I^*(X) \geq cin_I(S)$.

Proposition 8.1 (**Interval convex hull** [112, 146])**.** Let (X, I) be an interval space and $cin_I(S)$ the closure iteration number of an arbitrary subset $S \in \mathcal{P}(X)$. The convex hull of S, $co(S)$, fulfils $co(S) = cl_I^k(S)$ for any $k \geq cin_I(S)$; if S is already convex, then $cin_I(S) = 0$. If X is an infinite set, $cin_I(S)$ can be infinitely large and so $co(S) = cl_I^\infty(S)$ in general.

Proposition 8.1 (adapted from Pelayo's Proposition 2.1 [112] and Van de Vel's Proposition 4.1.2 [146]) states that a convex hull of a set in an interval space can be constructed recursively by applying the recursive pre-hull operator for at least as many times as the closure iteration number of such set (Definition 8.10). Because the support function of crossovers in $\mathcal{RP}$ or $\mathcal{I}_{fin}$ are interval operators, such crossovers have corresponding recursive pre-hull operators.

Then, in terms of crossovers, generating an interval convex hull via their recursive pre-hull operator entails generating all possible descendants of a given set of parents by applying the crossover operator to all parent pairs and then to their offspring recursively as Stadler, Wagner and others note [18, 51, 131]. Indeed, the set of all descendants obtained by recursively applying a crossover on a set of parents is their *full dynastic span*, which may be formalised via pre-hull operators (compare Definition 8.10 with Mitavskiy's Definition A.5 [98] and Radcliffe's Definition 37 [117]). In short, the interval convex hull of a set of parents yields the full dynastic span of such parents (compare Proposition 8.1 with Mitavskiy's Proposition A.4 [98]).

The following three examples illustrate the relationship between a crossover's support function and the interval convex hull it induces on a set of parents. For visual clarity, all examples use a reference Hamming metric space, even though recombination P-structures and finite interval spaces need not be associated with metric spaces. Appendices A.1–A.4 provide computer programs that helped to prepare the examples.

[2]Some authors call it *geodetic iteration number* when I is a geodesic interval [112].

Example 1: One-point Crossover vs Uniform Crossover

Let us compare first the uniform recombination P-structure $(\mathcal{H}_2^3, \textsc{Uniform})$ and one-point recombination P-structure $(\mathcal{H}_2^3, \textsc{OnePoint})$ corresponding to the support functions of uniform and one-point crossovers (Definitions 3.5 and 4.5) on three-dimensional binary Hamming sequences $\mathcal{H}_2^3$.

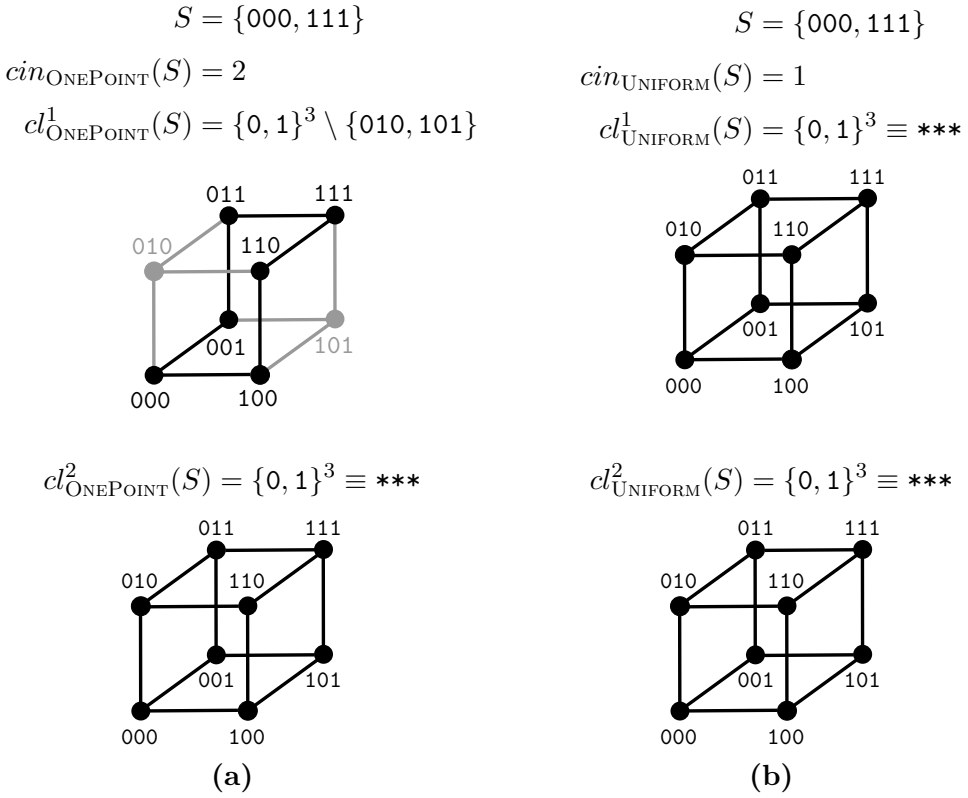


Figure 8.5. Comparison between the recursive pre-hull operators associated with the one-point recombination P-structure $(\mathcal{H}_2^3, \textsc{OnePoint})$ in Figure 8.5a and the uniform recombination P-structure $(\mathcal{H}_2^3, \textsc{Uniform})$ in Figure 8.5b for three-dimensional binary Hamming sequences $\mathcal{H}_2^3$ and given set of parents $S = \{000, 111\}$. Uniform crossover requires only one recursive application of the pre-hull operator to generate the convex hull of S, that is $cl^1_{\textsc{Uniform}}(S) = \{0,1\}^3$ or schema $***$, whereas one-point crossover requires two recursive applications to generate the convex hull of S, that is $cl^2_{\textsc{OnePoint}}(S) = \{0,1\}^3$.

Let the set of parents $S = \{000, 111\}$. Then, applying recursively the pre-hull operator for $\textsc{Uniform}$ on S (Figure 8.5b):

$$cl^0_{\textsc{Uniform}}(\{000, 111\}) = \{000, 111\} \ ,$$
$$cl^1_{\textsc{Uniform}}(\{000, 111\}) = \cup \{\textsc{Uniform}(x, y) \mid x, y \in cl^0_{\textsc{Uniform}}(\{000, 111\})\} = \{0,1\}^3 \ ,$$
$$cl^2_{\textsc{Uniform}}(\{000, 111\}) = \cup \{\textsc{Uniform}(x, y) \mid x, y \in cl^1_{\textsc{Uniform}}(\{000, 111\})\} = \{0,1\}^3 \ .$$

Hence the convex hull of S is $co(S) = cl^1_{\text{UNIFORM}}(S) = \{0,1\}^3$ or schema $***$, and the closure iteration number is $cin_{\text{UNIFORM}}(S) = 1$ since $cl^2_{\text{UNIFORM}}(S) = cl^1_{\text{UNIFORM}}(S)$. Analogously, for one-point crossover (Figure 8.5a):

$$
\begin{aligned}
cl^0_{\text{ONEPOINT}}(\{000,111\}) &= \{000,111\} \ , \\
cl^1_{\text{ONEPOINT}}(\{000,111\}) &= \cup \{\text{ONEPOINT}(x,y) \mid x,y \in cl^0_{\text{ONEPOINT}}(\{000,111\})\} \\
&= \{000,001,011,100,110,111\} = \{0,1\}^3 \setminus \{010,101\} \ , \\
cl^2_{\text{ONEPOINT}}(\{000,111\}) &= \cup \{\text{ONEPOINT}(x,y) \mid x,y \in cl^1_{\text{ONEPOINT}}(\{000,111\})\} \\
&= \{0,1\}^3 \ , \\
cl^3_{\text{ONEPOINT}}(\{000,111\}) &= \cup \{\text{ONEPOINT}(x,y) \mid x,y \in cl^2_{\text{ONEPOINT}}(\{000,111\})\} \\
&= \{0,1\}^3 \ .
\end{aligned}
$$

The convex hull is again $co(S) = cl^2_{\text{ONEPOINT}}(S) = \{0,1\}^3$. Therefore, $\{0,1\}^3$ is the set of all descendants (or full dynastic span) of parents $S = \{000,111\}$ under uniform and one-point crossovers. Clearly, both crossovers induce a geodesic convexity for Hamming spaces since they are geometric crossovers under Hamming distance. The pre-hull operator of one-point crossover requires, however, one extra iteration compared with uniform crossover. In general, one-point crossover requires the same or more iterations than uniform crossover to achieve the same convex hull [18, 51].

Example 2: Uniform Crossover vs All-Hamming-paths Crossover

This example illustrates how a non-geodesic interval convexity differs from a geodesic one by comparing uniform crossover and the non-geometric all-Hamming-paths crossover (Definition 5.4). Let $(\mathcal{H}^3_2, \text{UNIFORM})$ be a uniform recombination P-structure. Let also $(\mathcal{H}^3_2, \text{ALLPATHS}(H(3,2)))$ be an all-Hamming-paths recombination P-structure, abbreviated $(\mathcal{H}^3_2, \text{ALLPATHS})$, where $H(3,2)$ is a three-dimensional binary Hamming graph whose vertex set are three-dimensional binary Hamming sequences $\mathcal{H}^3_2$. Now fix the set of parents $S = \{010,110,100\}$ and apply recursively the pre-hull operator for UNIFORM on S:

$$
\begin{aligned}
cl^0_{\text{UNIFORM}}(\{010,110,100\}) &= \{010,110,100\} \ , \\
cl^1_{\text{UNIFORM}}(\{010,110,100\}) &= \cup \{\text{UNIFORM}(x,y) \mid x,y \in cl^0_{\text{UNIFORM}}(\{010,110,100\})\} \\
&= \{000,010,100,110\} = \{0,1\}^2 \times \{0\} \ , \\
cl^2_{\text{UNIFORM}}(\{010,110,100\}) &= \cup \{\text{UNIFORM}(x,y) \mid x,y \in cl^1_{\text{UNIFORM}}(\{010,110,100\})\} \\
&= \{0,1\}^2 \times \{0\} \ .
\end{aligned}
$$

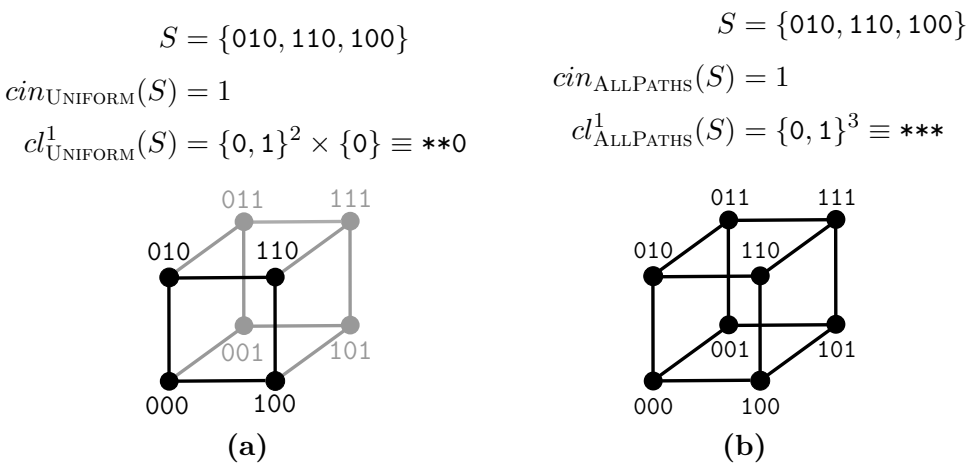


Figure 8.6. Comparison between the recursive pre-hull operators associated with the uniform recombination P-structure $(\mathcal{H}_2^3, \text{UNIFORM})$ in Figure 8.6a and the all-Hamming-paths recombination P-structure $(\mathcal{H}_2^3, \text{ALLPATHS})$ in Figure 8.6b for three-dimensional binary Hamming sequences $\mathcal{H}_2^3$ and given set of parents $S = \{010, 110, 100\}$. The convex hull of S induced by uniform crossover is $cl^1_{\text{UNIFORM}}(S) = \{0,1\}^2 \times \{0\}$ or schema $**0$, and the convex hull of S induced by all-Hamming-paths is $cl^1_{\text{ALLPATHS}}(S) = \{0,1\}^3$ or schema $***$. Both crossovers require only one recursive application of their pre-hull operators to generate the convex hull of S.

Therefore, the convex hull of S induced by UNIFORM crossover (Figure 8.6a) is $co(S) = cl^1_{\text{UNIFORM}}(S) = \{0,1\}^2 \times \{0\}$, or equally the schema $**0$, since further recursive applications $cl^2_{\text{UNIFORM}}(S)$, $cl^3_{\text{UNIFORM}}(S)$, etc., yield $**0$ again. However, ALLPATHS crossover induces a different convex hull:

$$
\begin{aligned}
cl^0_{\text{ALLPATHS}}(\{010, 110, 100\}) &= \{010, 110, 100\} \ , \\
cl^1_{\text{ALLPATHS}}(\{010, 110, 100\}) &= \cup \{\text{ALLPATHS}(x, y) \mid x, y \in cl^0_{\text{ALLPATHS}}(\{010, 110, 100\})\} \\
&= \{0,1\}^3 \ , \\
cl^2_{\text{ALLPATHS}}(\{010, 110, 100\}) &= \cup \{\text{ALLPATHS}(x, y) \mid x, y \in cl^1_{\text{ALLPATHS}}(\{010, 110, 100\})\} \\
&= \{0,1\}^3 \ .
\end{aligned}
$$

The convex hull of S induced by ALLPATHS crossover is $co(S) = cl^1_{\text{ALLPATHS}}(S) = \{0,1\}^3$ or the schema $***$ (Figure 8.6b). This is due to the degenerate behaviour of ALLPATHS (Lemma 5.1): for identical parents, ALLPATHS outputs such parent, and for distinct parents, it outputs the Hamming graph's vertex set entirely. As a result, the only possible convex sets induced by ALLPATHS are: the empty set $\emptyset$, singleton sets $\{v\}$ where $v \in V(H(3,2))$, and the entire vertex set $V(H(3,2))$ of the Hamming graph $H(3,2)$ parametrising ALLPATHS. Any other set S, such as $\{010, 110, 100\}$, is not convex: $\text{ALLPATHS}(x, y) \not\subseteq S$, $\forall x, y \in S$.

Observe that both UNIFORM and ALLPATHS induce an interval convex space, yet

only the former is geodesic. $C = \{0,1\}^2 \times \{0\}$ is geodesically convex for UNIFORM since it is a (complete) geometric crossover under Hamming distance, so $\forall x, y \in C$: $\text{UNIFORM}(x, y) = [x, y]_{d_H} \subseteq C$; however, because ALLPATHS is non-geometric some of its offspring lie outside C: $\text{ALLPATHS}(010, 000) \ni 111 \notin C$, for $010, 000 \in C$.

Example 3: Hamming Ball-mutation Segment Crossover

For the convexities in the previous examples, the convex hull of a single point would be exactly that point as one could intuitively expect based on traditional Euclidean convexity. Quite the opposite is true for the non-geodesic interval convexity induced by the ball-mutation segment crossover (Definition 8.2): the convex hull of a single point is the entire space. This example illustrates it.

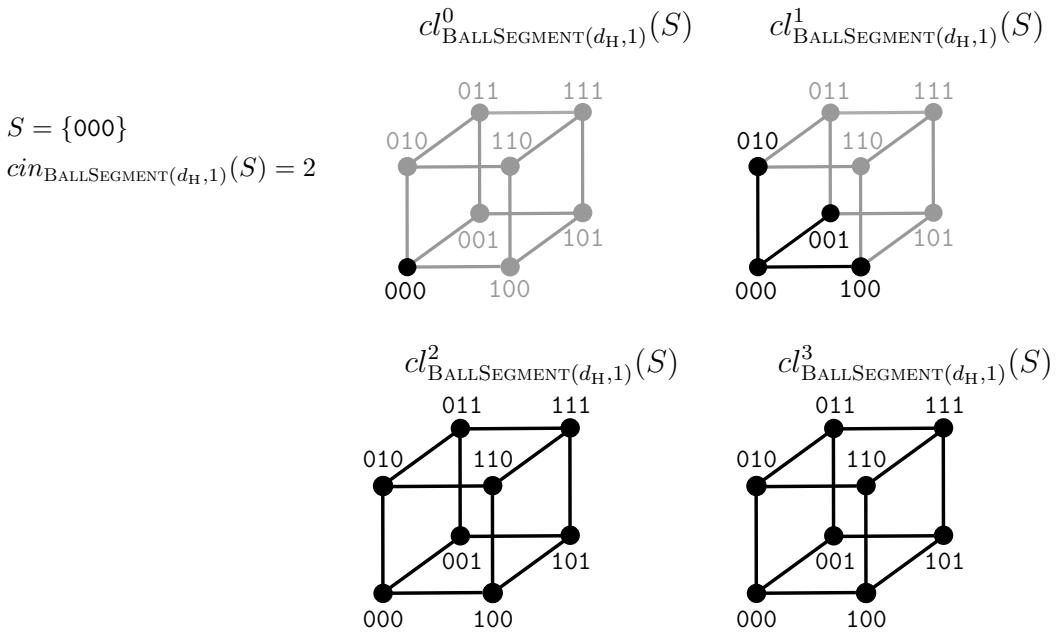


Figure 8.7. Recursive application of the pre-hull operator for the finite interval space $(\mathcal{H}_2^3, \text{BALLSEGMENT}(d_H, 1))$ associated with the ball-mutation segment crossover with radius one $\text{BALLSEGMENT}(d_H, 1)$ on three-dimensional binary Hamming sequences $\mathcal{H}_2^3$ under Hamming distance d_H. After two recursive applications of the pre-hull operator, the convex hull of $\{000\}$ induced by $\text{BALLSEGMENT}(d_H, 1)$ is $cl^2_{\text{BALLSEGMENT}(d_H,1)}(\{000\}) = \{0,1\}^3$ or schema $***$.

Let $(\mathcal{H}_2^3, \text{BALLSEGMENT}(d_H, 1))$ be a finite interval space for the ball-mutation segment crossover, parametrised with Hamming distance d_H and radius one, on three-dimensional binary Hamming sequences $\mathcal{H}_2^3$. Fix a set of parents $S = \{000\}$. Applying the pre-hull operator of $\text{BALLSEGMENT}(d_H, 1)$ recursively on S yields the

convex hull $cl^2_{\text{BallSegment}(d_{\text{H}},1)}(S) = \{\texttt{0},\texttt{1}\}^3$ or schema $\texttt{***}$ (Figure 8.7):

$$
\begin{aligned}
&cl^0_{\text{BallSegment}(d_{\text{H}},1)}(\{\texttt{000}\}) = \{\texttt{000}\} \ , \\
&cl^1_{\text{BallSegment}(d_{\text{H}},1)}(\{\texttt{000}\}) \\
&\quad = \cup \{\text{BallSegment}(d_{\text{H}},1)(x,y) \mid x,y \in cl^0_{\text{BallSegment}(d_{\text{H}},1)}(\{\texttt{000}\})\} \\
&\quad = \{\texttt{000},\texttt{001},\texttt{010},\texttt{100}\} \ , \\
&cl^2_{\text{BallSegment}(d_{\text{H}},1)}(\{\texttt{000}\}) \\
&\quad = \cup \{\text{BallSegment}(d_{\text{H}},1)(x,y) \mid x,y \in cl^1_{\text{BallSegment}(d_{\text{H}},1)}(\{\texttt{000}\})\} \\
&\quad = \{\texttt{0},\texttt{1}\}^3 \ , \quad (\text{note } \texttt{111} \in \text{BallSegment}(d_{\text{H}},1)(\texttt{001},\texttt{010})) \\
&cl^3_{\text{BallSegment}(d_{\text{H}},1)}(\{\texttt{000}\}) \\
&\quad = \cup \{\text{BallSegment}(d_{\text{H}},1)(x,y) \mid x,y \in cl^2_{\text{BallSegment}(d_{\text{H}},1)}(\{\texttt{000}\})\} \\
&\quad = \{\texttt{0},\texttt{1}\}^3 \ .
\end{aligned}
$$

In fact, regardless of which set S is chosen, as long as it is non-empty, the convex hull is $\{\texttt{0},\texttt{1}\}^3$. In this example, the closure iteration number will be at worst $cin_{\text{BallSegment}(d_{\text{H}},1)}(S) = 2$ when S is singleton (Figure 8.7); increasing the number of points in S, can only reduce its closure iteration number. The reason being that $\emptyset$ and $\{\texttt{0},\texttt{1}\}^3$ are the only convex sets for the interval convexity induced by the ball-mutation segment crossover. Due to the mutation component, any other set S distinct from $\emptyset$ and $\{\texttt{0},\texttt{1}\}^3$ such as $\{\texttt{000}\}$ is not convex as in Definition 8.7: $\text{BallSegment}(d_{\text{H}},1)(x,y) \not\subseteq S$, $\forall x,y \in S$.

To summarise, the key points of the three previous examples are:

- In general, interval spaces (X,I) require an infinite number of iterations to construct a convex hull using the recursive pre-hull operator. For finite interval spaces it can be achieved in finitely many iterations since X is finite.
- Some non-geometric crossovers (e.g. all-Hamming-paths and ball-mutation segment) induce 'degenerate' interval convexities because applying them (repeatedly) on a set of parents may produce all individuals in a given space.
- Different crossovers may induce different abstract interval convexities with different notions of convex hull even if the crossovers are defined on the same set of candidate solutions (e.g. binary Hamming sequences).
- Interval convex hulls may be geodesic or non-geodesic.

8.4 Abstract Behaviour of Formal Interval EAs

This section proves that the abstract behaviour of formal interval EAs (Definition 8.4) or any subclass thereof, including recombination P-structure EAs (Definition 8.5), is abstract interval convex evolutionary search as shown in Theorem 8.7 and its Corollary 8.1. In particular, for geometric recombination P-structure EAs, Corollary 8.2 shows abstract interval convex evolutionary search reduces to abstract geodesically convex evolutionary search (Proposition 6.1). That is, abstract interval convex evolutionary search may be geodesically convex or non-geodesically convex depending on the underlying crossover class; see Table 8.3 below.

If the crossover belongs to the class of geometric recombination P-structures $\mathcal{RP}$-geometric $= \mathcal{GX} \cap \mathcal{RP}$, then its interval convexity is geodesic as for geometric crossovers, and offspring will always lie within a geodesic convex hull of parents (Proposition 6.2), leading to abstract geodesically convex evolutionary search. By contrast, if the crossover is non-geometric and belongs either to $\mathcal{RP}$ or $\mathcal{I}_{fin}$, then its interval convexity can be non-geodesic, and offspring can lie within a possibly non-geodesic interval convex hull of parents (Lemma 8.2), leading to abstract non-geodesically convex evolutionary search.

Crossover class	EA class	Interval convexity	Abstract interval convex evolutionary search
$\mathcal{I}_{fin}$	Formal interval EA	Finite interval convexity	Non-geodesically convex
$\mathcal{RP}$	Recombination P-structure EA	Recombination P-structure convexity	
$\mathcal{RP}$-geometric	Geometric recomb. P-struct. EA	Geodesic convexity	Geodesically convex

Table 8.3. Main types of abstract interval convex evolutionary search associated with crossover classes, EA classes and abstract convexities.

Lemma 8.2. Selection, crossover and replacement population operators of a formal interval EA are convex population operators for any finite interval space (X, I).

Proof. First, note that by Definition 8.10 and Proposition 8.1 the maximum closure iteration number $m = cin_I^*(X)$ is finite $m < \infty$ for finite interval spaces. So the recursive pre-hull operator always produces a convex hull after m iterations $co(S) = cl_I^m(S)$ regardless of $S \subseteq X$ since $m \geq cin_I(S)$ the closure iteration number of S. To simplify notation, multi-sets (i.e. populations) are treated in terms of their support sets. The rest of this proof is similar to Moraglio's [101] proof of Proposition 6.2 for geometric-crossover EAs.

Selection. Let $P' = s(P)$ be a population resulting from a selection population operator s on an input population P as defined in Section 8.2.2. If $P' \subseteq co(P)$, then s is a convex population operator. By definition of s, $P' \subseteq P$, and by Kuratowski's axiom of extensivity in Definition 8.9, $P \subseteq co(P)$. Therefore, $P' \subseteq P \subseteq co(P)$.

Crossover. Let $P' = c(P)$ be a population resulting from a crossover population operator c on an input population P as defined in Section 8.2.2. If $P' \subseteq co(P)$, then c is a convex population operator. By definition of c, all offspring $z \in P'$ fulfil $z \in I(x, y)$ for at least one pair of parents $x, y \in P$. By Definition 8.10, $I(x, y) \subseteq cl_I^m(\{x, y\}) \subseteq cl_I^m(P)$. By Proposition 8.1, $co(S) = cl_I^m(S)$ for any subset S. Therefore, $I(x, y) \subseteq co(\{x, y\}) \subseteq co(P)$, hence $P' \subseteq co(P)$.

Replacement. Let $P_3 = r(P_1, P_2)$ be a population resulting from a replacement population operator r on input populations P_1 and P_2 as defined in Section 8.2.2. If whenever $P_2 \subseteq co(P_1)$, $P_3 \subseteq co(P_1)$ holds, then r is a convex population operator. By definition of r, $P_3 \subseteq P_1 \cup P_2$. Moreover, $P_3 \subseteq co(P_3)$ by Kuratowski's axiom of extensivity, and $P_3 \subseteq P_1 \cup P_2 \implies co(P_3) \subseteq co(P_1 \cup P_2)$ by Kuratowski's axiom of isotony (Definition 8.9). So $P_3 \subseteq co(P_3) \subseteq co(P_1 \cup P_2)$. Now, suppose $P_2 \subseteq co(P_1)$, then:

$$\begin{aligned} co(P_1 \cup P_2) &\subseteq co(P_1 \cup co(P_1)) \\ &= co(co(P_1)) && \text{using Kuratowski's axiom of extensivity} \\ &= co(P_1) && \text{by Kuratowski's axiom of idempotency.} \end{aligned}$$

Therefore, $P_3 \subseteq co(P_3) \subseteq co(P_1 \cup P_2) \subseteq co(P_1)$. □

Theorem 8.7 (Abstract interval convex evolutionary search [49]). Let (X, I) be any finite interval space and $P_t \subseteq X$ a population at generation $t \geq 0$, with P_0 being the initial population. Then, there exists a finite maximum closure iteration number $m = cin_I^*(X)$ such that $cl_I^m(P_0) \supseteq cl_I^m(P_1) \supseteq \cdots \supseteq cl_I^m(P_t) \supseteq cl_I^m(P_{t+1})$, for any formal interval EA repeating the cycle of population operators: selection, crossover and replacement.

Proof. We need to prove that such maximum closure iteration number exists and the nested chain of inclusions is formed:

(a) By Proposition 8.1, for any population P_t there is a natural number $k \geq cin(P_t)$ such that $co(P_t) = cl_I^k(P_t)$. By Definition 8.10, the maximum closure iteration

number $m = cin_I^*(X)$ fulfils $m \geq k = cin(P_t)$ since all populations P_t are subsets of X and $m < \infty$ since (X, I) is finite. Hence $\forall P_t : co(P_t) = cl_I^m(P_t)$.

(b) From (a) there is a finite maximum closure iteration number m such that $\forall P_t : co(P_t) = cl_I^m(P_t)$. By Lemma 8.2, selection s, crossover c and replacement r are convex population operations, so their sequential application on a parent population P_t produces an offspring population $P_{t+1} = r\Big(P_t, c(s(P_t))\Big)$ where $P_{t+1} \subseteq co(P_t)$ since $c(s(P_t)) \subseteq co(P_t)$. Therefore, $co(P_{t+1}) \subseteq co(co(P_t)) = co(P_t)$ by Kuratowski's axioms of isotony and idempotency (Definition 8.9), so the nested chain of inclusions follows by induction on t since t was chosen arbitrarily. □

Restricting the underlying crossover of formal interval EAs from the class of finite interval operators $\mathcal{I}_{fin}$ to recombination P-structures $\mathcal{RP}$ leads to recombination P-structure EAs (Section 8.2.2) whose behaviour is also abstract interval convex evolutionary search (Corollary 8.1).

Corollary 8.1 (**Abstract interval convex evolutionary search** [49])**.** Let $(X, \mathcal{R})$ be any recombination P-structure and $P_t \subseteq X$ a population at generation $t \geq 0$, with P_0 being the initial population. Then, there exists a finite maximum closure iteration number $m = cin_{\mathcal{R}}^*(X)$ such that $cl_{\mathcal{R}}^m(P_0) \supseteq cl_{\mathcal{R}}^m(P_1) \supseteq \cdots \supseteq cl_{\mathcal{R}}^m(P_t) \supseteq cl_{\mathcal{R}}^m(P_{t+1})$, for any recombination P-structure EA repeating the cycle of population operators: selection, crossover and replacement.

Proof. It follows immediately from Theorem 8.7 and that any recombination P-structure EA is a formal interval EA by Definition 8.5. □

If recombination P-structures $\mathcal{RP}$ are further restricted to geometric recombination P-structures $\mathcal{RP}$-geometric, then the corresponding recombination P-structure EA becomes a geometric-crossover EA. In such case, abstract interval convex evolutionary search and abstract geodesically convex evolutionary search become equivalent (see Corollary 8.2 and Figure 8.8 below).

Corollary 8.2 (**Abstract interval convex evolutionary search ≡ abstract geodesically convex evolutionary search** [49])**.** Any recombination P-structure EA over a geometric recombination P-structure does abstract geodesically convex evolutionary search. Also, any geometric-crossover EA over a geometric recombination P-structure does abstract interval convex evolutionary search. So abstract interval convex evolutionary search and abstract geodesically convex evolutionary search are equivalent when both are restricted to geometric recombination P-structures.

Proof. Any crossover in the class $\mathcal{RP}$-geometric $= \mathcal{GX} \cap \mathcal{RP}$ is both a geometric crossover and a (geometric) recombination P-structure by Theorem 5.15. Hence by Definitions 6.3 and 8.5, any recombination P-structure EA using geometric recombination P-structures is a geometric-crossover EA; and conversely, any geometric-crossover EA using a geometric recombination P-structure is a recombination P-structure EA. Therefore, by Corollary 8.1 any geometric-crossover EA using geometric recombination P-structures does abstract interval convex evolutionary search; and, by Proposition 6.1 any recombination P-structure EA using geometric recombination P-structures does abstract geodesically convex evolutionary search. □

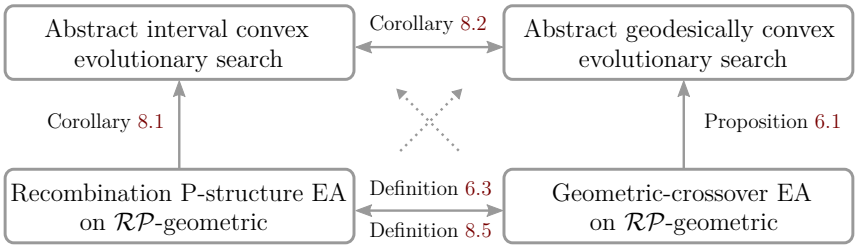


Figure 8.8. Equivalence between abstract interval convex evolutionary search and abstract geodesically convex evolutionary search for recombination P-structure EAs and geometric-crossover EAs on geometric recombination P-structures $\mathcal{RP}$-geometric. Bidirectional arrows mean equivalence; unidirectional arrows mean conformity.

Remark 8.1 (Theorem 8.7 and Corollaries 8.1–8.2)**.** Abstract interval convex evolutionary search does not necessarily generalise abstract geodesically convex evolutionary search of geometric-crossover EAs for geometric crossovers that are not finite interval operators, namely crossovers in the class $\mathcal{GX} \cap \overline{\mathcal{I}_{fin}}$.

According to Theorem 8.7 and Corollaries 8.1–8.2, any formal interval EA does a form of abstract interval convex evolutionary search, whether geodesically convex or not, regardless of: a specific representation of solutions, problem (fitness function), crossover in the class $\mathcal{I}_{fin}$ (possibly using mutation) without assumptions on specific probability distributions over offspring, selection or replacement mechanisms, and whether the population size varies or not through generations. Theorem 8.7 is true not just for crossovers covered by recombination P-structures $\mathcal{RP}$ but any kind of crossover as long as it is symmetric and returns parents in the offspring set, namely finite interval operators $\mathcal{I}_{fin}$. Corollary 8.2 proves that is exactly equivalent to abstract geodesically convex evolutionary search when restricting to crossovers in $\mathcal{RP}$-geometric, but what about other (non-geometric) crossovers in $\mathcal{RP}$ or $\mathcal{I}_{fin}$? The following Section 8.4.1 exemplifies both abstract geodesically convex and non-

geodesically convex evolutionary search to grasp how 'analogous' they really are, postponing a discussion to Section 8.5.

8.4.1 Illustrating Formal Interval EAs Convergence

An abstract interval convex evolutionary search performed by a formal interval EA after $t \geq 0$ generations, in a finite interval space (X, I) with maximum closure iteration number $m = cin_I^*(X)$, is described by the nested chain of interval convex hulls $cl_I^m(P_0) \supseteq \cdots \supseteq cl_I^m(P_t)$ corresponding to a specific sequence of populations $(P_i)_{i\in\{0,\ldots,t\}}$ that such formal interval EA has generated (Theorem 8.7). We can distinguish the following three principal cases of the nested inclusion chain:

(a) Strictly decreasing: $cl_I^m(P_0) \supset cl_I^m(P_1) \supset \cdots \supset cl_I^m(P_t)$.

(b) Weakly decreasing, so the nested chain of interval convex hulls decreases at some point. That is, there exist $i, j \in \{0, \ldots, t\}$ where $i < j$ such that $cl_I^m(P_i) \supset cl_I^m(P_j)$.

(c) Non-decreasing: $cl_I^m(P_0) = cl_I^m(P_1) = \cdots = cl_I^m(P_t)$.

Observe that for each generation i, the population P_i fulfils $P_i \subseteq cl_I^m(P_i)$ due to Kuratowski's extensivity axiom (Definition 8.9) provided that $co(P_i) = cl_I^m(P_i)$ due to Proposition 8.1. Thus a nested chain strictly or weakly decreasing over generations forces populations to confine themselves to increasingly more specific regions of the interval space (X, I). In this sense, three 'coarse-grained' types of convergence for abstract interval convex evolutionary search are distinguished (Definition 8.11). For the sole illustrative purposes of this section is not necessary to make Definition 8.11 more precise nor distinguish other possible cases than the ones above.

Definition 8.11 (**Formal interval EA: coarse-grained convergence**)**.** The abstract interval convex evolutionary search of a formal interval EA for a particular sequence of populations is said to be: (a) *strictly convergent*, if the nested chain of interval convex hulls is strictly decreasing; (b) *weakly convergent*, if the nested chain of interval convex hulls is weakly decreasing; or, (c) *stationary* or *non-convergent*, if if the nested chain of interval convex hulls is non-decreasing.

The *actual search* of a formal interval EA is described by the sequence of populations generated, instead of their nested chain of interval convex hulls. Hereinafter, if a sequence of populations is said to *converge*, it means that the number of copies of specific candidate solutions or individuals increase so populations become more homogeneous over generations [4].

Definition 8.11 allows us to easily compare how, for a given population sequence, different crossovers in the class $\mathcal{I}_{fin}$ of finite interval operators lead to different types of convergence by comparing the nested chain of interval convex hulls that such crossovers induce. This links explicitly the extended classification of crossovers (Section 8.2.1) with their associated interval convexities (Section 8.3) and convergence of formal interval EAs from a qualitative viewpoint. Recall from Section 8.3 that certain non-geometric crossovers in $\mathcal{I}_{fin}$, like all-Hamming-paths or Hamming ball-mutation segment, induce degenerate convexities because repeatedly applying them on a set of parents can produce offspring anywhere in the search space of solutions. Yet any formal interval EA based on them does abstract interval convex evolutionary search due to Theorem 8.7. The purpose of the following examples summarised in Table 8.4 is to illustrate that, even if the actual sequence of populations converges to a specific individual, the abstract convex evolutionary search can be stationary or non-convergent. The abstract behaviour of a formal interval EA may not always reflect the specific behaviour of the populations it generates.

EA	Selection	Crossover	Mutation	Replacement	Abstract behaviour
Geometric recombination P-structure EA	*	UNIFORM	None	Generational	Figure 8.9a
Recombination P-structure EA	*	BALLS(d_{H})	None	Generational	Figure 8.9b
Recombination P-structure EA	*	ALLPATHS($H(4,2)$)	None	Generational	Figure 8.9c
Formal interval EA	*	BALLSEGMENT($d_{\mathrm{H}}, 1$)	Yes†	Generational	Figure 8.9d

* Fixed but unspecified
† BALLSEGMENT($d_{\mathrm{H}}, 1$) equates to uniform crossover followed by single bit-flip mutation (see Figure 8.2b)

Table 8.4. Examples of formal interval EAs, given by main population operators, and associated abstract behaviours. Crossover operators are: uniform (UNIFORM), intersecting-Hamming-balls (BALLS(d_{H})), all-Hamming-paths (ALLPATHS($H(4,2)$)), and Hamming ball-mutation segment with radius one (BALLSEGMENT($d_{\mathrm{H}}, 1$)); where $H(4,2)$ is a four-dimensional binary Hamming graph and d_{H} the Hamming distance.

The examples in Figure 8.9 consider distinct variants of a formal interval EA (Table 8.4) that only differ in their respective crossover and all the following are fixed: (a) individuals represented by binary Hamming sequences of length four, (b) no mutation (except for the mutation implicit in ball-mutation segment crossover), (c) generational replacement, and (d) the population size is four. To facilitate their comparison, the examples assume all these formal interval EAs can generate the same sequence of populations in Figure 8.9e via some selection mechanism that is fixed but left unspecified since any fitness related aspect is not a concern for the examples' purpose.

	P_0	P_1	P_2	P_3	P_4
$k = cin_{\text{UNIFORM}}(P_i)$	1	1	1	0	0
$cl^k_{\text{UNIFORM}}(P_i)$	$\underset{(****)}{\{0,1\}^4}$ $\supset$	$\underset{(0***)}{\{0\} \times \{0,1\}^3}$ $\supset$	$\underset{(00**)}{\{0\}^2 \times \{0,1\}^2}$ $\supset$	$\underset{(001*)}{\{0\}^2 \times \{1\} \times \{0,1\}}$ $\supset$	$\{0011\}$

(a) Uniform: geodesically convex, strictly convergent

	P_0	P_1	P_2	P_3	P_4
$k = cin_{\text{BALLS}(d_{\text{H}})}(P_i)$	2	2	2	0	0
$cl^k_{\text{BALLS}(d_{\text{H}})}(P_i)$	$\underset{(****)}{\{0,1\}^4}$ $=$	$\underset{(****)}{\{0,1\}^4}$ $=$	$\underset{(****)}{\{0,1\}^4}$ $\supset$	$\underset{(001*)}{\{0\}^2 \times \{1\} \times \{0,1\}}$ $\supset$	$\{0011\}$

(b) Intersecting-Hamming-balls: non-geodesically convex, weakly convergent

	P_0	P_1	P_2	P_3	P_4
$k = cin_{\text{ALLPATHS}(H(4,2))}(P_i)$	1	1	1	1	0
$cl^k_{\text{ALLPATHS}(H(4,2))}(P_i)$	$\underset{(****)}{\{0,1\}^4}$ $=$	$\underset{(****)}{\{0,1\}^4}$ $=$	$\underset{(****)}{\{0,1\}^4}$ $=$	$\underset{(****)}{\{0,1\}^4}$ $\supset$	$\{0011\}$

(c) All-Hamming-paths: non-geodesically convex, weakly convergent

	P_0	P_1	P_2	P_3	P_4
$k = cin_{\text{BALLSEGMENT}(d_{\text{H}},1)}(P_i)$	1	1	2	2	3
$cl^k_{\text{BALLSEGMENT}(d_{\text{H}},1)}(P_i)$	$\underset{(****)}{\{0,1\}^4}$ $=$	$\underset{(****)}{\{0,1\}^4}$ $=$	$\underset{(****)}{\{0,1\}^4}$ $=$	$\underset{(****)}{\{0,1\}^4}$ $=$	$\underset{(****)}{\{0,1\}^4}$

(d) Hamming ball-mutation segment: non-geodesically convex, stationary

P_0: $\{(1\ 1\ 0\ 1), (0\ 0\ 1\ 0), (0\ 0\ 1\ 0), (0\ 1\ 0\ 0)\}$

P_1: $\{(0\ 0\ 1\ 0), (0\ 1\ 0\ 1), (0\ 0\ 0\ 1), (0\ 0\ 0\ 0)\}$

P_2: $\{(0\ 0\ 1\ 0), (0\ 0\ 1\ 1), (0\ 0\ 0\ 1), (0\ 0\ 1\ 0)\}$

P_3: $\{(0\ 0\ 1\ 0), (0\ 0\ 1\ 1), (0\ 0\ 1\ 1), (0\ 0\ 1\ 1)\}$

P_4: $\{(0\ 0\ 1\ 1), (0\ 0\ 1\ 1), (0\ 0\ 1\ 1), (0\ 0\ 1\ 1)\}$

(e) Populations

Figure 8.9. Abstract interval convex evolutionary search (Figures 8.9a–8.9d) induced by each formal interval EA variant in Table 8.4 on an initial population P_0 during four generations (Figure 8.9e). For each population P_i, each example indicates the corresponding interval convex hull (or schema) produced by the pre-hull operator $cl^k_{\cdot}(P_i)$ and closure iteration number k for crossovers: uniform (UNIFORM), intersecting-Hamming-balls (BALLS(d_{H})), all-Hamming-paths (ALLPATHS($H(4,2)$)), and Hamming ball-mutation segment (BALLSEGMENT($d_{\text{H}}, 1$)). The closure iteration numbers are obtained with the help of computer programs in Appendices A.1–A.4, except for all-Hamming-paths which is trivially one for any set with at least two different elements.

Figure 8.9 shows abstract interval convex evolutionary search, from strictly convergent in Figure 8.9a to stationary in Figure 8.9d, for various crossovers in all three main crossover classes ($\mathcal{GX}$, $\mathcal{RP}$ and $\mathcal{I}_{fin}$) that comprehend distinctive features of geometric crossovers, monotonic recombination P-structures and highly local crossovers (see Definitions 3.3, 4.9 and 5.7 respectively):

- Uniform crossover: geometric, monotonic and highly local.
- Intersecting-Hamming-balls crossover: non-geometric, non-monotonic and highly local.
- All-Hamming-paths crossover: non-geometric and non-monotonic.
- Hamming ball-mutation segment crossover: non-geometric, non-monotonic and uses mutation.

Out of these, uniform crossover, precisely the only geometric crossover, is the only one inducing abstract interval convex evolutionary search that strictly converges (Figure 8.9a) according to Definition 8.11. Observe the nested chain of interval convex hulls indicates a succession of increasingly specific schemas that reflect how populations converge towards `0011`. However, that is not the case for the non-geometric crossovers above; their interval convex hulls degenerate to $\{0, 1\}^4$ or schema `****` for most (or all) of the populations, so the corresponding nested chains no longer reflect the actual progress of populations towards `0011`. For instance, the intersecting-Hamming-balls and all-Hamming-paths crossovers induce abstract interval convex evolutionary search that weakly converges (Figures 8.9b–8.9c), but it is stationary for most generations. Only when the populations have (practically) converged to `0011`, do their interval convex hulls become more specific than schema `****`. The last and least convergent of all examples is in Figure 8.9d, where the Hamming ball-mutation segment crossover induces the stationary type of abstract interval convex evolutionary search: nothing relevant can be said about the actual population behaviour from the nested chain of interval convex hulls, except that the schema `****` matches all populations.

Changing the populations in Figure 8.9e does not fundamentally change the types of abstract interval convex evolutionary search in Figures 8.9a–8.9d because the it does not fundamentally change the interval convexities and convex hulls induced by the crossovers. The interval convex hulls induced by Hamming ball-mutation segment, all-Hamming-paths and intersecting-Hamming-balls crossovers would be `****` as long as the populations are respectively: non-empty, contain two individuals

that differ in at least one position, and contain two individuals that differ in at least two positions.

Other Examples of Abstract Interval Convex Evolutionary Search

Although Figure 8.9 exemplifies all coarse-grain types of convergence for abstract interval convex evolutionary search, it does not cover all cases for geodesic and non-geodesic convexities neither all crossovers in the class $\mathcal{I}_{fin}$ of finite interval operators. These other cases may not occur in practice, they may be analogous, or it may be hard to find non-trivial examples. For instance:

- *Identity crossover.* By Definition 4.4, Id is a fix-point operator $\text{Id}(x, y) = \{x, y\}$ for all parents $x, y \in S \subseteq X$ and finite set X, so any offspring set $\text{Id}(x, y)$ is interval convex for any such $x, y \in S$: $\text{Id}(x, y) \subseteq S$. Trivially, the interval convex hull is $cl^0_{\text{Id}}(S) = cl^1_{\text{Id}}(S) = S$. In the absence of mutation and selection, any formal interval EA only using Id generates trivial sequences of populations $P_0, P_1, \ldots, P_t$ identical to the initial population P_0; that may not be the case for other crossovers. So $cl^0_{\text{Id}}(P_0) = \cdots = cl^0_{\text{Id}}(P_0)$ is the only possible nested chain since $cl^0_{\text{Id}}(P_0) = P_0$. Therefore, the abstract interval convex evolutionary search is stationary according to Definition 8.11.

- *One-point crossover.* Given the same sequence of populations, OnePoint crossover induces the same abstract interval convex evolutionary search as Uniform crossover (i.e. strictly convergent for the example in Figure 8.9a). The reason is their respective interval convex hulls $cl^k_{\text{OnePoint}}(S)$ and $cl^{k'}_{\text{Uniform}}(S)$ coincide for any set S of Hamming sequences, even if their closure iteration numbers $k = cin_{\text{OnePoint}}(S)$ and $k' = cin_{\text{Uniform}}(S)$ do not (Section 8.3).

- *Uniform crossover.* Uniform need not always induce abstract interval convex evolutionary search that is strictly convergent as in Figure 8.9a. If all populations in Figure 8.9e had all individuals differing in all positions (i.e. $\forall i \in \{0, \ldots, 4\}\, \forall x, y \in P_i : d_{\text{H}}(x, y) = 4$), then the nested chain of interval convex hulls would be $\ast\ast\ast\ast = \cdots = \ast\ast\ast\ast$ as for $\text{BallSegment}(d_{\text{H}}, 1)$ crossover in Figure 8.9d. In practice, such sequence of populations may be unlikely; assuming no mutation and some level of elitism via selection, populations would begin to converge, so individuals would not differ in all positions. In this sense, Uniform is unlikely to induce the stationary type of abstract interval convex evolutionary search, as opposed to crossovers like $\text{BallSegment}(d_{\text{H}}, 1)$ which always induces it whenever populations are non-empty.

- *Symmetric Davis's order crossover.* Except for trivial populations (e.g. empty or containing identical individuals), all-Hamming-paths, intersecting-Hamming-balls and Hamming ball-mutation segment crossovers never induce the strictly convergent type of abstract interval convex evolutionary search. By contrast, sym-DAVIS crossover (Definition 5.2) is the only known non-geometric crossover in the class $\mathcal{I}_{fin}$ which compares to geometric crossovers like UNIFORM because, for some non-trivial sequences of populations, it can induce the strictly convergent type of abstract interval non-geodesically convex evolutionary search as in Figure 8.10 for the permutation space $\mathcal{S}_X$ of elements in $X = \{1, 2, 3, 4, 5\}$.

	P_0	P_1	P_2	P_3
$k = cin_{\text{sym-DAVIS}}(P_i)$	2	0	0	0
$cl^k_{\text{sym-DAVIS}}(P_i)$	$\mathcal{S}_{\{1,2,3,5\}} \times \{4\} \supset$	$\mathcal{S}_{\{1,2,3\}} \times \{5\} \times \{4\} \supset$	$\{13254, 31254\} \supset$	$\{31254\}$

(a) Symmetric Davis's order: non-geodesically convex, strictly convergent

$$\underset{P_0}{\left\{\begin{matrix}(2&1&3&5&4),\\(1&2&3&5&4),\\(1&3&5&2&4),\end{matrix}\right\}} \quad \underset{P_1}{\left\{\begin{matrix}(2&1&3&5&4),\\(1&3&2&5&4),\\(3&1&2&5&4),\end{matrix}\right\}} \quad \underset{P_2}{\left\{\begin{matrix}(1&3&2&5&4),\\(3&1&2&5&4),\\(3&1&2&5&4),\end{matrix}\right\}} \quad \underset{P_3}{\left\{\begin{matrix}(3&1&2&5&4),\\(3&1&2&5&4),\\(3&1&2&5&4),\end{matrix}\right\}}$$

(b) Populations

Figure 8.10. Abstract interval convex evolutionary search (Figure 8.10a) induced on an initial population P_0 during three generations (Figure 8.10b) by a formal interval EA with symmetric Davis's order crossover, no mutation and generational replacement. For each population P_i, $cl^k_{\text{sym-DAVIS}}(P_i)$ indicates the corresponding interval convex hull with closure iteration number $k = cin_{\text{sym-DAVIS}}(P_i)$ for symmetric Davis's order crossover. (see Appendix A.2 for the computer program used to prepare the example.)

8.4.2 Interpreting Abstract Convex Evolutionary Search as Nested Generalised Schemas or Invariant Subsets

The examples in Figure 8.9 illustrate that abstract interval convex evolutionary search in finite interval spaces based on binary Hamming sequences can be described by Holland's schemas [65]. The nested chain of populations' convex hulls corresponds to a succession of schemas, where the schema of an offspring population is either equal or more specific than its parent population. In binary Hamming metric spaces, schemas have a dual nature: they may refer to geodesically convex sets of binary Hamming sequences or syntactic patterns representing those sets (see Section 6.2.2 for more details).

However, abstract interval convex evolutionary search (Theorem 8.7) applies to any finite interval space, not just those defined on binary Hamming sequences. Fortunately, there is a representation-independent notion of schemas, called *generalised schemas* [98, 99], which generalises Holland's when viewed as subsets of solutions rather than syntactic patterns representing them. Indeed, generalised schemas are also called *invariant subsets* precisely because they are the subsets of a space which are *closed* or *invariant* under a given operator, that is Definition 8.12.

Definition 8.12 (**Generalised schema** [98, 99])**.** Let an arbitrary non-empty set X. Then, $S \subseteq X$ is a *generalised schema* or *invariant subset* with respect to a given operator defined on X, if for any input parents $P \subseteq S$ the operator outputs all offspring as a subset of S.

Based on Definition 8.12, Mitavskiy's Proposition A.1 [98] shows that the family of all generalised schemas or invariant subsets in some arbitrary space actually constitutes a family of convex sets of a corresponding aligned space (Definition 8.8); interval convex spaces being a particular case of them (Definition 8.7) as Figure 8.4 showed in Section 8.3. Thus, the (interval) convex hull of a given set is the most specific generalised schema or smallest invariant subset that includes the given set, leading immediately to the next Corollary 8.3 of Theorem 8.7.

Corollary 8.3 (**Interpretation via generalised schema**)**.** Let (X, I) be any finite interval space. Let also $cl_I^m(P_0) \supseteq cl_I^m(P_1) \supseteq \cdots \supseteq cl_I^m(P_t) \supseteq cl_I^m(P_{t+1})$ be the nested chain of interval convex hulls on populations $P_t \subseteq X$ produced by a formal interval EA through generations $t \geq 0$, where $m = cin_I^*(X)$ is the maximum closure iteration number. Then, any $cl_I^m(P_t)$ is a generalised schema or invariant subset with respect to $r\Big(P_t, c(s(P_t))\Big)$, that is the sequential application of selection s, crossover c and replacement r population operators of the formal interval EA.

Proof. Recall from Theorem 8.7 that $P_{t+1} = r\Big(P_t, c(s(P_t))\Big) \subseteq co(P_t) = cl_I^m(P_t)$ holds because of Proposition 8.1 and because s, c and r are convex population operators (Lemma 8.2). Since any interval convex hull $cl_I^m(P_t)$ is a convex set, it is also a generalised schema or invariant subset as in Definition 8.12 due to Mitavskiy's Proposition A.1 [98]. □

Remark 8.2 (Alternative proof of Corollary 8.3)**.** To prove $co(P_t)$ is an invariant subset with respect to the sequential application of s, c and r, we have to prove such application on any population $P'_t \subseteq co(P_t)$ yields an offspring population P'_{t+1} such that $P'_{t+1} \subseteq co(P_t)$. That is, to prove $P'_{t+1} = r\Big(P'_t, c(s(P'_t))\Big) \subseteq co(P_t)$ holds $\forall P'_t \subseteq co(P_t)$. Suppose $P'_t \subseteq co(P_t)$. Then, $co(P'_t) \subseteq co(co(P_t)) = co(P_t)$ by

Kuratowski's axioms of isotony and idempotency (Definition 8.9). Because s and c are convex population operators, $c(s(P'_t)) \subseteq co(P'_t)$. Since r is also a convex population operator and $c(s(P'_t)) \subseteq co(P'_t)$, it follows: $r\Big(P'_t, c(s(P'_t))\Big) \subseteq co(P'_t)$. Therefore, $P'_{t+1} = r\Big(P'_t, c(s(P'_t))\Big) \subseteq co(P'_t) \subseteq co(P_t)$.

Then, Corollary 8.3 proves abstract interval convex evolutionary search (Theorem 8.7) has an alternative interpretation via generalised schemas or invariant subsets. Any formal interval EA produces a sequence of populations corresponding to a nested inclusion chain of generalised schemas or invariant subsets associated with a geodesic or non-geodesic interval convexity, regardless of: a specific representation of solutions, problem (fitness function), crossover in the class $\mathcal{I}_{fin}$ (possibly using mutation) without assumptions on specific probability distributions over offspring, selection or replacement mechanisms, and whether the population size varies or not through generations.

It must be noted that, unlike Holland's schemas, generalised schemas as in Definition 8.12 may not always correspond to a syntactic pattern; and, if one does, it may not suitably describe allelic similarities of offspring produced by the crossover at hand. (That is, offspring sharing specific alleles at specific positions.) The following example shows how generalised schemas, or interval convex sets, induced by the symmetric Davis's order crossover (Definition 5.3) correspond to a syntactic pattern for permutations enforcing specific symbols (i.e. alleles) must appear at fixed positions. This pattern is Goldberg's *absolute ordering schema* [53] under a slightly different notation. However, the symmetric Davis's order crossover, like the original (non-symmetrised) Davis's order crossover (Definition 5.2) [31], respects the relative ordering of parents alleles in the offspring but not necessarily their absolute ordering for all offspring.

Example Based on Symmetric Davis's Order Crossover for Permutations

Recall from Figure 8.10, in Section 8.4.1, the finite interval space $(\mathcal{S}_X, \text{sym-DAVIS})$ over the set of permutations (without repetitions) $\mathcal{S}_X$ of $X = \{1, 2, 3, 4, 5\}$ and sym-DAVIS the symmetric Davis's order crossover. The nested chain of interval convex hulls (Figure 8.10a) induced by sym-DAVIS for the populations in Figure 8.10b is:

$$(\mathcal{S}_{\{1,2,3,5\}} \times \{4\}) \supset (\mathcal{S}_{\{1,2,3\}} \times \{5\} \times \{4\}) \supset (\mathcal{S}_{\{1,3\}} \times \{2\} \times \{5\} \times \{4\}) \supset \{\texttt{31254}\} \ .$$

It can be regarded a nested chain of generalised schemas as follows.

Observe $\mathcal{S}_{\{1,2,3,5\}} \times \{4\}$ comprises only those permutations whose rightmost element is four, out of all possible in $\mathcal{S}_X$. Therefore, $\mathcal{S}_{\{1,2,3,5\}} \times \{4\}$, or a permutation

of the form $s_1s_2s_3s_4\texttt{4}$, can be said to be a generalised schema where $\texttt{4}$ is fixed and the juxtaposed sequence of variables $s_1s_2s_3s_4$ acts as a 'wildcard' representing any permutation in $\mathcal{S}_{\{1,2,3,5\}}$. Likewise, $\mathcal{S}_{\{1,2,3\}} \times \{5\} \times \{4\}$ comprises only those permutations whose two rightmost elements are five and four, out of all possible in $\mathcal{S}_X$. Therefore, $\mathcal{S}_{\{1,2,3\}} \times \{5\} \times \{4\}$, or a permutation of the form $s_1s_2s_3\texttt{54}$, would be a generalised schema where $\texttt{5}$ and $\texttt{4}$ are fixed and $s_1s_2s_3$ is any permutation in $\mathcal{S}_{\{1,2,3\}}$. More precisely, $s_1s_2s_3s_4\texttt{4}$ and $s_1s_2s_3\texttt{54}$ are the smallest invariant subsets or generalised schemas induced by sym-Davis that include, respectively, populations P_0 and P_1 in Figure 8.10b.

Clearly, $\mathcal{S}_{\{1,2,3\}} \times \{5\} \times \{4\}$ is a subset of $\mathcal{S}_{\{1,2,3,5\}} \times \{4\}$, or, in other words, $s_1s_2s_3\texttt{54}$ is a generalised schema that is more specific than $s_1s_2s_3s_4\texttt{4}$. Following the same reasoning for the rest of the populations explains why the aforesaid nested chain of interval convex hulls may be regarded a nested chain of generalised schemas.

In contrast with Holland's schema, where a wildcard is a single symbol $*$ that always represents any of $\{0, 1, \ldots, q-1\}^n$ for n-dimensional q-ary ($q \geq 2$) Hamming sequences, here a wildcard has the form of a permutation that varies depending on the fixed symbols of the generalised schema. The wildcard for $s_1s_2s_3\texttt{54}$ is a permutation of length three $s_1s_2s_3 \in \mathcal{S}_{\{1,2,3\}}$ where $\texttt{5}$ and $\texttt{4}$ cannot be used in $s_1s_2s_3$; for instance, $s_1\texttt{4}s_3\texttt{54}$ and $s_1\texttt{51254}$ are not generalised schemas because this example considers permutations without repetitions and do not match any permutation in $\mathcal{S}_{\{1,2,3\}} \times \{5\} \times \{4\}$.

A Note on Monotonic Recombination P-structures vs Generalised Schemas

Interestingly, the nested chain of generalised schemas induced by a formal interval EA may be regarded a generalisation of the invariance present in monotonic recombination P-structures $(X, \mathcal{R})$, where $\mathcal{R}(u, v) \subseteq \mathcal{R}(x, y)$ for any parents $x, y \in X$ and offspring $u, v \in \mathcal{R}(x, y)$. Offspring sets $\mathcal{R}(x, y)$ of monotonic recombination P-structures are, by Definition 4.9, generalised schemas (Lemma 8.3) and always coincide with the interval convex hull of parents (Theorem 8.8).

Lemma 8.3. Let $(X, \mathcal{R})$ be any monotonic recombination P-structure and arbitrary $x, y \in X$. Then, $\mathcal{R}(x, y)$ is a generalised schema with respect to $\mathcal{R}$.

Proof. To prove that the offspring set $\mathcal{R}(x, y)$, for arbitrary parents $x, y \in X$, is a generalised schema with respect to $\mathcal{R}$, we have to prove that recombining any possible subset of offspring $S \subseteq \mathcal{R}(x, y)$ results in grandchildren that belong to $\mathcal{R}(x, y)$, that is $\forall u, v \in S \subseteq \mathcal{R}(x, y) : \mathcal{R}(u, v) \subseteq \mathcal{R}(x, y)$. It follows immediately from Definition 4.9 of monotonic recombination P-structures. □

Theorem 8.8. Let $(X, \mathcal{R})$ be any monotonic recombination P-structure and arbitrary $x, y \in X$. Then, $\mathcal{R}(x, y) = cl^k_{\mathcal{R}}(\{x, y\}) = co(\{x, y\})$ and $k = cin_{\mathcal{R}}(\{x, y\}) \leq 1$.

Proof. From Definition 8.10 of recursive pre-hull operator:

$$\begin{aligned} cl^0_{\mathcal{R}}(\{x, y\}) &= \{x, y\} \ , \\ cl^1_{\mathcal{R}}(\{x, y\}) &= \cup \{\mathcal{R}(x', y') \mid x', y' \in \{x, y\}\} \\ &= \cup \{\mathcal{R}(x, x), \mathcal{R}(y, y), \mathcal{R}(x, y), \mathcal{R}(y, x)\} \\ &= \mathcal{R}(x, y) \ , \end{aligned} \tag{8.1}$$

where the last equality (Equation 8.1) follows from symmetry and null-recombination axioms of recombination P-structures (Definition 4.3). Now, either $\{x, y\}$ is interval convex or it is not.

- Case 1. Assume it is interval convex, so $\forall x', y' \in \{x, y\} : \mathcal{R}(x', y') \subseteq \{x, y\}$. Then, $cl^1_{\mathcal{R}}(\{x, y\}) = \mathcal{R}(x, y) = \{x, y\} = cl^0_{\mathcal{R}}(\{x, y\})$ for the null-recombination axiom $\{x, y\} \subseteq \mathcal{R}(x, y)$ and the above assumption implying $\mathcal{R}(x, y) \subseteq \{x, y\}$.
- Case 2. Assume it is not interval convex, so $\forall x', y' \in \{x, y\} : \mathcal{R}(x', y') \not\subseteq \{x, y\}$. Then, continuing with the recursive pre-hull operator:

$$\begin{aligned} cl^2_{\mathcal{R}}(\{x, y\}) &= \cup \{\mathcal{R}(x', y') \mid x', y' \in cl^1_{\mathcal{R}}(\{x, y\})\} \\ &= \cup \{\mathcal{R}(x', y') \mid x', y' \in \mathcal{R}(x, y)\} \\ &= \mathcal{R}(x, y) \\ &= cl^1_{\mathcal{R}}(\{x, y\}) \ , \end{aligned} \tag{8.2}$$

  where the penultimate equality in Equation 8.2 follows from: the axiom of null-recombination $\{x, y\} \subseteq \mathcal{R}(x, y)$, and $\mathcal{R}$ being a monotonic recombination P-structure or generalised schema $\mathcal{R}(x', y') \subseteq \mathcal{R}(x, y)$ (Lemma 8.3).

In either case $\mathcal{R}(x, y) = cl^k_{\mathcal{R}}(\{x, y\}) = co(\{x, y\})$ and the closure iteration number is $k \leq 1$. □

8.5 Discussion

This chapter showed the class of formal interval EAs do a generalised form of abstract convex evolutionary search and can use some geometric or non-geometric crossovers as well as mutation, effectively unifying geometric-crossover EAs and recombination

P-structure random walks by asking a fundamental two-fold question. Is there any class of EAs, other than geometric-crossover EAs, based on recombination P-structures and whose behaviour is analogous to abstract convex evolutionary search? Is it still true, if such EAs use mutation? (Question 8.1.)

The discussion of Question 8.1 and formal interval EAs is split into: the extended classification of crossovers (Section 8.5.1), a qualitative comparison between formal interval EAs and geometric-crossover EAs (Section 8.5.2) as well as formal interval EAs and recombination P-structure random walks (Section 8.5.3), and abstract interval convex evolutionary search (Section 8.5.4).

8.5.1 Extended Classification of Crossovers

Extending the original crossover classification in Chapter 5 with the class $\mathcal{I}_{fin}$ of finite interval operators may seem not very useful. Assuming that is relatively easy to design crossovers that are symmetric and return parents in the offspring set, then one may regard $\mathcal{I}_{fin}$ overgeneralised inasmuch as it includes virtually any crossover. Strictly speaking that is not the case; not all crossovers are provably interval operators whether finiteness is assumed or not: the asymmetric one-point and Davis's order crossovers are two examples due to their asymmetry (see Section 8.2.1).

Nevertheless, this chapter does not claim that the extended crossover classification is any more practical than the original one. The purpose was to investigate in line with Moraglio and Stephens [100, 141] how classifying crossovers helps us formally understand the abstract behaviour of EA classes defined over crossover classes and, in particular, address Question 8.1. This chapter showed that neither the axioms of finite interval operators, nor recombination P-structures $\mathcal{RP}$ proposed by Stadler and others [51, 139], are strong enough to ensure abstract interval convex evolutionary search is always equivalent to abstract geodesically convex evolutionary search induced by geometric crossovers $\mathcal{GX}$ (see Section 8.4). An exception is the subclass $\mathcal{RP}$-geometric (Corollary 8.2). Also, the symmetric Davis's order crossover is neither geometric nor a recombination P-structure, yet the example in Figure 8.10 shows it induces strictly convergent type of abstract interval convex evolutionary search as geometric crossovers like uniform crossover (see Figure 8.9a).

8.5.2 Formal Interval EAs vs Geometric-crossover EAs

Unlike geometric-crossover EAs, EAs with mutation or non-geometric crossovers provably do not do abstract geodesically convex evolutionary search [101]. Then, how formal interval EAs possibly do any form of abstract convex evolutionary search

if they use non-geometric crossovers and mutation? It suffices to look again at the crossovers distinguishing geometric-crossover EAs and formal interval EAs.

Geometric-crossover EAs always require a crossover ξ that is geometric for at least one distance d of some metric space (X, d). By Definition 3.3, geometric crossovers always produce offspring within geodesic intervals between given parents. Therefore, geometric crossovers are always associated with a geodesic convexity $(X, \mathcal{C})$ since $\xi(x, y) \subseteq [x, y]_d \subseteq C$, for any geodesically convex set $C \in \mathcal{C}$ and parents $x, y \in C$. That is fundamentally why geometric-crossover EAs can produce offspring populations within the geodesic convex hull of parents, hence abstract geodesically convex evolutionary search (Proposition 6.1).

By contrast, formal interval EAs require a crossover associated with a finite interval space but not a distance nor have to be geometric: they may use any crossover in the class $\mathcal{I}_{fin}$ of finite interval operators (Figure 8.1). Some finite interval operators are not geometric crossovers (e.g. all-Hamming-paths and Hamming ball-mutation segment) and so do not produce offspring within the geodesic convex hull of parents (Section 8.3). However, that does not imply such crossovers are not convex with respect to any convexity! The key from Section 8.3 is every crossover associated with a finite interval space induces its own notion of interval convexity and convex hull (i.e. intrinsic to the crossover). This applies to any crossover in $\mathcal{I}_{fin}$, whether it is geometric or not, whether it uses mutation or not. Then, formal interval EAs do abstract interval convex evolutionary search (Theorem 8.7) because, if the interval convexity specific to the crossover is taken into account, they produce offspring populations within the interval convex hull of parent populations by Lemma 8.2.

A counter-intuitive fact about formal interval EAs is that they comprehend also certain EAs with mutation and no crossover, which is impossible for geometric-crossover EAs by Definition 6.3. The class $\mathcal{I}_{fin}$ of finite interval operators is so general that it includes forms of 'crossover' with mutation. Like ball-mutation segment crossover (Definition 8.2), one could define the *union-balls* 'crossover' parametrised with a distance d and ball radius $r \in \mathbb{R}_{>0}$ in a finite metric space (X, d):

$$\textsc{UnionBalls}(d, r)(x, y) \stackrel{\text{def}}{=} \bar{B}_d(x, r) \cup \bar{B}_d(y, r) \ ,$$

that for parents $x, y \in X$ it returns as offspring all neighbours (alongside parents) within the respective d-metric closed balls centred at each parent. For instance, the offspring of parents $\mathtt{001}$ and $\mathtt{100}$ under Hamming distance d_{H} and ball radius one are: $\textsc{UnionBalls}(d_{\mathrm{H}}, 1)(\mathtt{001}, \mathtt{100}) = \{\mathtt{001}, \mathtt{000}, \mathtt{011}, \mathtt{101}\} \cup \{\mathtt{100}, \mathtt{000}, \mathtt{101}, \mathtt{110}\}$. Effectively, $\textsc{UnionBalls}(d_{\mathrm{H}}, 1)$ applies a single-bit flip mutation to both parents.

UnionBalls(d, r) trivially fulfils the symmetry and extensivity axioms of finite interval spaces, that is UnionBalls$(d, r) \in \mathcal{I}_{fin}$, and it is neither a geometric crossover nor recombination P-structure for similar reasons to Theorem 8.1. Hence, according to Theorem 8.7, a formal interval EA using union-balls crossover does abstract interval convex evolutionary search even if union-balls is not really a crossover! This is counter-intuitive because, according to Moraglio [101], mutation operators are not geodesically convex and thus break the nested chain of geodesic convex hulls. Like ball-mutation segment crossover (Figure 8.9d), union-balls would also induce the stationary type of abstract interval convex evolutionary search (Definition 8.11).

The bottom line is that abstract geodesically convex evolutionary search of geometric-crossover EAs can be generalised to abstract interval convex evolutionary search of formal interval EAs. However, that involves unnatural forms of crossover with counter-intuitive interval convexities (e.g. union-balls and ball-mutation segment), which can lead to formal interval EAs whose abstract search is theoretically sound but does not tell anything relevant about the actual population behaviour.

8.5.3 Formal Interval EAs vs Recombination P-structure Random Walks

Until now, headless-chicken (HC) crossover random walks for recombination P-structures were the only recombination-based search model analysed in ELT (Section 7.3). As an alternative, this chapter proposes recombination P-structure EAs, a subclass of formal interval EAs, for the following reasons.

HC crossover random walks significantly differ from conventional EAs (compare Algorithms 6.1 and 7.1). At any time, a HC crossover random walk would evolve just one individual by applying a recombination P-structure to two parents, where one is given from a previous recombination and the other is randomly sampled from a prescribed probability distribution over the solution set. But no actual selection, crossover nor replacement as in EAs are involved. The only way a HC crossover random walk could simulate an EA is by prescribing a sequence of probability distributions that reflected the evolution of a population subject to some selection force. However, in that case it is pointless to use a HC crossover random walk to begin with. Furthermore, ELT made additional assumptions (Sections 7.3.2–7.3.3) restricting HC crossover random walks such as unrealistically assuming a uniform probability distribution over the entire solution set, or a string-based representation of individuals with none (or sufficiently low) statistical interdependence between values at any position in the string (i.e. between alleles at different loci).

By contrast, a recombination P-structure EA (Definition 8.5) does have true population dynamics, and it can use: any finite representation of individuals, selection, replacement, crossover (as long as it is a recombination P-structure), and population size (fixed or variable). Also, any probability distribution may be defined over offspring because it is transparent to the support sets of crossover; put differently, the recombination P-structure in a recombination P-structure EA makes no assumptions on an underlying probability distribution over offspring. In any case, the behaviour of a recombination P-structure EA is abstract interval convex evolutionary search (Corollary 8.1) or abstract geodesically convex evolutionary search (Corollary 8.2) when restricting to geometric recombination P-structures. Does it mean a recombination P-structure EA cannot degenerate to pure random search if degenerate crossovers like all-Hamming-paths are used? No, it is possible; however, HC crossover random walks do not offer much choice beyond pure random search.

8.5.4 Abstract Interval Convex Evolutionary Search

Formal interval EAs do abstract interval convex evolutionary search in finite interval spaces (Theorem 8.7), and it could extend to infinite interval spaces, in principle, if the assumption of finiteness is dropped. But, in doing so, abstract interval convex evolutionary search incurs a cost due to Proposition 8.1: the closure iteration number of a subset $S \subseteq X$ may be infinite. Consequently, for a sequence of populations $(P_i)_{i \in \{0,\ldots,t\}}$ in an infinite interval space (X, I), where $P_i \subseteq X$, the corresponding nested chain of interval convex hulls may be *indeterminate*: $cl_I^\infty(P_0) \supseteq \cdots \supseteq cl_I^\infty(P_t)$.

Unequal one-point crossover is one such infinite interval operator [131], which is expected since Theorem 8.2 proved it is not a finite interval operator yet trivially fulfils the axioms of interval spaces (Definition 4.2). Recalling the example from Table 8.1, it is easy to see why the interval convex hull induced by unequal one-point crossover on $\{\texttt{00}, \texttt{11}\}$, for instance, is not finite:

$$\underbrace{\{\texttt{00},\texttt{11}\}}_{cl^0_{\text{uneq-OnePoint}}(\{\texttt{00},\texttt{11}\})} \neq \underbrace{\{\texttt{0011},\ldots\}}_{cl^1_{\text{uneq-OnePoint}}(\{\texttt{00},\texttt{11}\})} \neq \underbrace{\{\texttt{00110011},\ldots\}}_{cl^2_{\text{uneq-OnePoint}}(\{\texttt{00},\texttt{11}\})} \neq \cdots$$

In other words, there is no natural number $k < \infty$ such that $cl^{k+1}_{\text{uneq-OnePoint}}(\{\texttt{00}, \texttt{11}\}) = cl^k_{\text{uneq-OnePoint}}(\{\texttt{00}, \texttt{11}\})$. The same applies to Koza's subtree swap crossover [86] for an analogous reason (see Figure 8.3). Therefore, it is unclear whether abstract interval convex evolutionary search applies to virtually all crossovers as discussed earlier in Section 8.5.1.

Finite interval operators $\mathcal{I}_{fin}$, or recombination P-structures $\mathcal{RP}$, neither guar-

antee that abstract interval convex evolutionary search conveys any information on whether the populations of a formal interval EA are converging (Section 8.4.1). Even if populations converge due to selection or other bias, a formal interval EA using non-geometric crossovers in $\mathcal{RP}$ or $\mathcal{I}_{fin}$, such as all-Hamming-paths or Hamming ball-mutation segment, potentially induces an abstract interval convex evolutionary search that is (mostly) stationary (Figure 8.9). This disagreement in convergence between the abstract search and the actual search of formal interval EAs is a consequence of Theorem 8.7 and two underlying reasons. First, the nested chain of populations' interval convex hulls is determined only by the populations and the crossover, hence independent of selection and replacement operators[3]. Secondly, the crossovers themselves can generate individuals anywhere in the search space when applied recursively to a set of parents (Section 8.3), thereby favouring *exploration* like mutation operators, which results in degenerate interval convexities and ultimately a stationary abstract search performed by the formal interval EA.

Then, it becomes necessary to consider crossover classes with more restrictive axioms than $\mathcal{I}_{fin}$ and $\mathcal{RP}$ to avoid the stationary type of abstract interval convex evolutionary search. Preferably, one may want to find the conditions where abstract interval convex evolutionary search converges while being beneficial on problems. The following observations[4] altogether indicate what those conditions may be:

- Abstract interval convex evolutionary search of a formal interval EA forms a nested chain of interval convex hulls all of which are invariant subsets or generalised schema (Definition 8.12) with respect to the sequential application of selection, crossover and replacement population operators (Theorem 8.3).

- Vose [148] argues that GAs are effective when they produce a nested sequence of invariant subsets (called *stable* in Vose's terms) and the fitness function of a given problem increases monotonically along those subsets. Mitavskiy and He [98, 99] provide a methodology to design EAs accordingly by matching the invariant subsets with the fitness level sets of the fitness function.

- Moraglio and Sudholt [104] prove a specific geometric-crossover EA finds the global optimum of the 'leading-ones' function in $\mathcal{O}(n \log n)$ fitness evaluations, where leading-ones is quasi-concave function whose fitness level sets form a nested sequence of schemas (i.e. invariant subsets).

[3]Interval operators are very general, so it may be possible to define more complex interval convex hulls that take into account selection, crossover and replacement.

[4]The motivation behind some of these observations is Holland's schema theorem [65]. This chapter does not attempt to generalise Holland's schema theorem.

- Although not all highly local crossovers are geometric (Corollary 5.2), geometric crossovers (including geometric recombination P-structures) are always highly local. Rothlauf and Thorhauer [124, 144], as well as Droste and Wiesmann [38], recommend high locality based on similar grounds underlying principled design of geometric crossovers in Chapter 3: highly local crossovers produce offspring relatively close to parents with respect to a distance and, if such distance is correlated with a problem's fitness function, it favours *exploitative* search in promising regions of the problem's search space.

- Monotonic recombination P-structures are the subclass of recombination P-structures that produce offspring sets which are both invariant (Lemma 8.3) and convex (Theorem 8.8). Such invariance in Mitavskiy's sense [98] is a generalisation of Radcliffe's crossover design principle of *respect* [117] deemed relevant for an exploitative genetic search: any *formae* (i.e. alleles) shared by parents must be shared by their offspring. Reformulating Radcliffe's crossover design principles in terms of geodesic convexity is possible according to Hofmeyr [64].

These observations point to two prime requirements for convergence and efficacy of abstract interval convex evolutionary search. First, restricting the possible offspring produced by crossover, so that parents and offspring are similar to each other to an extent. Secondly, designing crossovers that can exploit relevant features of the fitness function (i.e. problem knowledge) during the search.

These requirements are feasible for abstract interval convex evolutionary search by limiting to crossover subclasses $\mathcal{RP}$-geometric and $\mathcal{RP}$-monotonic of recombination P-structures, which seem to align well with the aforesaid quasi-convex and monotonically increasing functions. All crossovers in $\mathcal{RP}$-geometric are geometric crossovers and thus can exploit problem knowledge by carefully selecting a distance for the problem, and crossovers in $\mathcal{RP}$-monotonic can achieve it if invariant subsets correspond to fitness level sets. Also, $\mathcal{RP}$-geometric and $\mathcal{RP}$-monotonic clearly restrict the possible offspring to a geodesic interval between parents or an invariant subset given by parents, so they are unlikely to induce degenerate interval convexities and display mutation-like behaviour as other crossovers in $\mathcal{I}_{fin}$ or $\mathcal{RP}$ do. Exceptions are the degenerate all-Hamming-path crossover in $\mathcal{RP}$-monotonic and the non-degenerate symmetric Davis's order crossover in $\mathcal{I}_{fin}$ which indeed can produce invariant subsets (Section 8.4.2). An unexplored but interesting crossover class is the strict-size monotonic recombination P-structures found in Chapter 5 because they fulfil the convergence inbreeding property of geometric crossovers seen in Chapter 3.

8.6 Conclusion

Formally understanding the population behaviour of EAs or classes thereof is a long-standing challenge in EC, including the GF and ELT. Despite all progress, the foundations of the GF and ELT are not fully satisfactory in the following two senses: (a) the EA classes they cover, across problems and representations, are not as general as one may expect (e.g. not using populations or mutation, missing useful crossovers, etc.); and, (b) they lack a systematic approach to rigorously understand how different search operators (e.g. crossover and mutation) affect the behaviour of such EA classes.

My original major contribution is a qualitative framework built upon the crossover classification in Chapter 5 to address issues (a) and (b) mentioned above, allowing a systematic analysis of how different crossover classes can lead to possibly different abstract behaviours in a corresponding general EA class. That is a framework to axiomatically analyse and classify the population behaviour of formal interval EAs, an EA class which generalises and integrates geometric-crossover EAs from the GF (Chapter 6) and recombination P-structure random walks from ELT (Chapter 7). Three main aspects of this framework set it apart from the GF and ELT:

- Introducing a crossover class (i.e. finite interval operators) that includes: non-geometric crossovers besides geometric ones, all recombination P-structures, and other more general crossovers some of which can even act as mutation.
- Defining formal interval EAs based on finite interval operators, thus generalising geometric-crossover EAs (which cannot use mutation and non-geometric crossovers) and recombination P-structure random walks (which clearly are not population-based EAs and can only use recombination P-structures).
- Showing that formal interval EAs do abstract interval convex evolutionary search across problems and representations, which generalises the abstract convex evolutionary search of geometric-crossover EAs and may not always coincide with it. Unlike geometric-crossover EAs in the GF, the abstract behaviour of formal interval EAs can be geodesically or non-geodesically convex depending on the crossover used. Unlike recombination P-structure random walks, formal interval EAs provide ELT with an EA model that has true population dynamics rather than headless-chicken walk dynamics.

Thus formal interval EAs can be more exploitative or explorative depending on the underlying crossover and whether mutation occurs or not. Generalising

abstract (geodesically) convex evolutionary search, which the GF only conceived for geometric-crossover EAs, is therefore possible as this chapter showed. But it comes at a cost: certain non-geometric crossovers and mutation can sometimes lead to degenerate forms of abstract interval convex evolutionary search that is no longer useful to describe the actual behaviour of populations. For this reason, it is necessary to characterise precisely the degenerate and non-degenerate kinds of abstract convexity associated with crossover classes, besides the abstract behaviours induced by them. The framework here presented can help us accomplish both.

Part III

Fitness Landscapes

Abstract

This third and last part develops upon Part I and Part II to unify the general classes of fitness landscapes proposed by the GF and ELT, as revised in background Chapters 9 and 10. Fitness landscapes in the GF are generalised across problems and representations as functional forms parametrised by: a metric space, providing a search space and its structure, and a fitness function defined on such metric space. This approach not only allows geometric-crossover EAs to be instantiated to a given fitness landscape by specifying the same metric distance, but also allows various classes of abstract convex landscapes to be defined as generalisations of traditional convex functions. ELT provides a general framework, based on spectral graph theory, to carry a Fourier-like decomposition analysis of landscapes. Of special interest are the general class of elementary landscapes based on mutation search spaces, whose fitness function happens to be an eigenfunction of an associated graph Laplacian matrix, present in several classical combinatorial problems and real-world problems. An analogous class of elementary landscapes can be defined based on recombination P-structures, or crossover search spaces, via recombination P-structure Laplacians. For certain recombination P-structures, ELT shows that if a fitness function is elementary then it is also recombination P-structure elementary and vice versa.

The contribution of this third part is presented in Chapter 11 focusing on the two-fold research question (3) from the literature review in Chapter 2: what class of combinatorial landscapes is shared by abstract convex landscapes and elementary landscapes? How does the discrete Laplacian operator corresponding to the shared landscape class relate to the difficulty of problems associated with it? To address these questions, Chapter 11 first identifies certain conditions to conceive abstract convex elementary landscapes precisely as a class of combinatorial landscapes that are abstract convex, as in the GF, and (recombination P-structure) elementary, as in ELT. To characterise them, various examples of pseudo-Boolean fitness functions are classified. This provably reveals the existence of abstract convex elementary landscapes: one-max functions (with or without certain bounded perturbations) are given as examples. Then, a more general characterisation reveals all (quasi-)convex or (quasi-)concave elementary landscapes correspond to graph Laplacian eigenvalues of order less than two, providing a direct and 'sampling-free' way to classify such abstract convex elementary landscapes and possibly others as well. Interestingly, the GF showed already geometric-crossover EAs in expectation can exponentially outperform random search on subclasses of quasi-concave (or quasi-concave) landscapes; whereas ELT showed order-one elementary landscapes are smooth landscapes.

Chapter 9

Abstract Convex Landscapes in the Geometric Framework

This chapter revises background material for subsequent chapters. It summarises and clarifies key ideas about classes of abstract convex fitness landscapes where geometric-crossover EAs, introduced in Chapter 6, may outperform pure random search. This chapter principally supports the contributions in Chapter 11 to find if such classes comprehend landscapes of real-world or NP-complete problems.

9.1 Matching Abstract Evolutionary Search and Abstract Fitness Landscapes for Performance

Geometric-crossover EAs (Definition 6.3) always do abstract geodesically convex evolutionary search (Proposition 6.1), irrespective of the optimisation problem and representation of solutions, on any search space defined by a metric space. But Proposition 6.1 itself does not state anything at all about geometric-crossover EAs performance [100]. Indeed, due to NFL theorems [68, 155], any EA would perform as well (or bad) as any other EA, including pure random search, if their performance is averaged across all possible optimisation problems whose objective function has a fixed discrete domain and co-domain.

Nevertheless, NFL does not prohibit that certain EAs outperform others when a restricted subset of problems is considered, or when EAs are specially designed for certain problems. This prompts GF to find those problems where geometric-crossover EAs provably outperform pure random search (at least), for GF would be a futile theory of EAs otherwise, by examining conditions of their corresponding *fitness landscapes* that abstract geodesically convex evolutionary search can exploit

profitably. In particular, GF proposes various classes of 'globally convex' fitness landscapes where geometric-crossover EAs may outperform pure random search [101]. This chapter presents those fitness landscape classes. Their relation with geometric-crossover EAs performance [104] is described only tangentially since this thesis does not analyse geometric-crossover EAs performance. Sections 9.1.1–9.1.2 overview first how fitness landscapes and geometric-crossover EAs relate one another, and what is the main motivation behind globally convex landscapes.

9.1.1 Fitness Landscapes and Geometric-crossover EAs

A *fitness function* is an assignment of numerical values, typically real numbers, to candidate solutions of an optimisation problem to quantify their optimality. Defining a fitness function over solutions with an associated search space structure, for example a metric space, constitutes a *fitness landscape*.

Definition 9.1 (**Fitness landscape**)**.** A *fitness landscape* is a triplet (X, d, f) where an arbitrary real-valued function $f : X \to \mathbb{R}$, the *fitness function*, is defined on an arbitrary metric space (X, d) with a set X and metric $d : X \times X \to \mathbb{R}_{\geq 0}$. A fitness landscape is called *flat* whenever f is a constant function.

Remark 9.1. Definition 9.1 assumes fitness functions are known closed-form mathematical expressions. Also, for simplicity, no distinction is made between fitness function and objective function, nor between encoded solutions (i.e. genotypes) and formal solutions (i.e. phenotypes).

$$\mathscr{F} = (X, d, f) \xrightarrow{\text{input}} \text{GX-EA}(\mathscr{F}) \xrightarrow{\text{run}} \text{ACS}(\mathscr{F}) \xrightarrow{\text{result}} \text{ASP}(\mathscr{F})$$

Figure 9.1. Causal relations between a fitness landscape $\mathscr{F}$, geometric-crossover EA (GX-EA), abstract convex evolutionary search (ACS) and the resulting abstract search performance (ASP). ASP is a general runtime expression parametrised with certain characteristics of the fitness landscape $\mathscr{F}$ (e.g. dimensionality, cardinality, distance, etc.).

Definition 9.1 describes generic fitness landscapes, not tied to a predetermined specific problem nor representation, in the same sense geometric-crossover EAs are abstract specifications of conventional EAs (Algorithm 6.1) without mutation. That is, fitness landscapes and geometric-crossover EAs are functional objects parametrised by a fixed but unspecified metric. Such abstraction has two main consequences [101]:

- Turns fitness landscapes and geometric-crossover EAs into separate but related entities. Definition 9.1 does not enforce specific geometric-crossover EAs on

a given fitness landscape (thus separate). Yet specific instances of geometric-crossover EAs can be defined on a landscape by choosing the same distance and specifying their population search operators in Section 6.1 (thus related).

- Facilitates a systematic understanding of fitness landscapes and geometric-crossover EAs because their functional forms can be automatically instantiated on any specific problem and search space using these as parameters. This is illustrated in Figure 9.1 through causal relations [101]. First, a geometric-crossover EA is instantiated by functional substitution of its components with an input fitness landscape; then, running that specific geometric-crossover EA produces a specific geodesically convex search on the input fitness landscape, which results in a specific performance for that landscape and EA.

The traditional viewpoint 'one operator one landscape' [73] conceives EAs searching not in one but multiple fitness landscapes because, in this view, each specific search operator defines its own fitness landscape: changing the search operator changes the landscape, hence one operator one landscape. So there would be mutation landscapes, crossover landscapes, selection landscapes, etc. GF fundamentally challenges the former view since different geometric-crossover EAs can search within the same fitness landscape, implying 'one landscape many operators' instead.

9.1.2 Why Globally Convex Fitness Landscapes?

Convex functions in continuous spaces can be optimised efficiently and effectively [12, 40]. The same is true of certain generalised convex functions in discrete spaces. For example, TSP, weighted bipartite matching and flow-shop scheduling problems have specific instances, provably solvable in polynomial time, where their objective function is *submodular* [11, 15][40, ch. 2] sharing certain properties with generalised forms of convexity [79, 90, 111, 132]. The potential performance benefits for EAs due to a 'globally convex trend' or 'big valley' in fitness landscapes [10] have been corroborated experimentally for some but not all instances of various combinatorial problems [46, 109]. This motivates GF to propose several fitness landscape classes [101] which to different extents have a globally convex trend, align well with abstract convex evolutionary search, and where geometric-crossover EAs may have exponentially better performance than pure random search [104]. Figure 9.2 summarises those classes. The following Sections 9.2–9.3 introduce them focusing on convex functions from the viewpoint of minimisation problems, but the same ideas apply to their counterparts based on concave functions for maximisation problems.

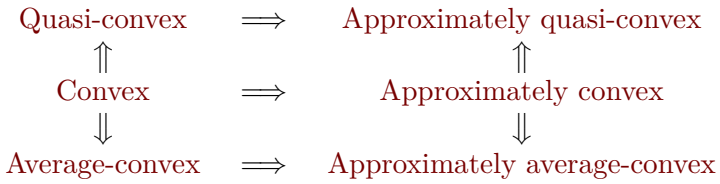


Figure 9.2. Overview of abstract convex fitness landscape classes and logical relationships: if a class implies another class, all members of the first are members of the second.

9.2 Towards General Convex Fitness Landscapes

Let us start with traditional *convex functions* in Euclidean spaces, or other vector spaces, and extend them to more general metric spaces.

Definition 9.2 (**Convex function** [12])**.** Let $f : C \to \mathbb{R}$ be a real-valued function defined on a convex subset $C \subseteq \mathbb{R}^n$. Then, f is *convex* if *Jensen's inequality* holds for all $x, y \in C$ and $\theta \in [0, 1]$:

$$f\Big(\theta x + (1-\theta)y\Big) \leq \theta f(x) + (1-\theta)f(y) \ . \tag{9.1}$$

If strict inequality holds in Equation 9.1, then f is *strictly convex*, whenever $x \neq y$ and $0 < \theta < 1$. Similarly, f is *concave* if $-f$ is convex; f is *strictly concave*, if $-f$ is strictly convex; and f is *affine*, if it is convex and concave.

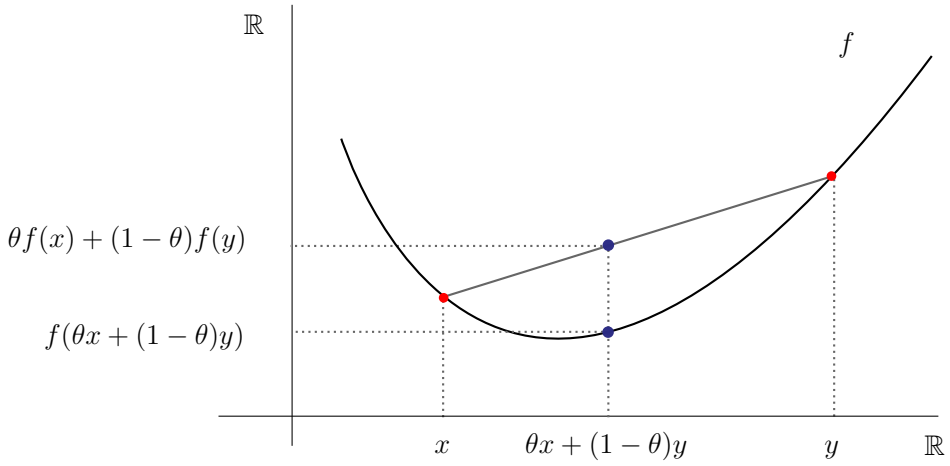


Figure 9.3. Visualisation of a strictly convex function using Jensen's inequality in a two-dimensional Euclidean metric space $(\mathbb{R}^2, d_2)$.

Remark 9.2 (Definition 9.2)**.** Jensen's inequality extends to convex combinations conforming the convex hull of a set. Let $A = \{a_1, \ldots, a_k\} \subset \mathbb{R}^n$ be a finite non-empty subset with $k > 0$. If f is a convex function, then $f\Big(\sum_{i=1}^{k} \theta_i a_i\Big) \leq \sum_{i=1}^{k} \theta_i f(a_i)$, where

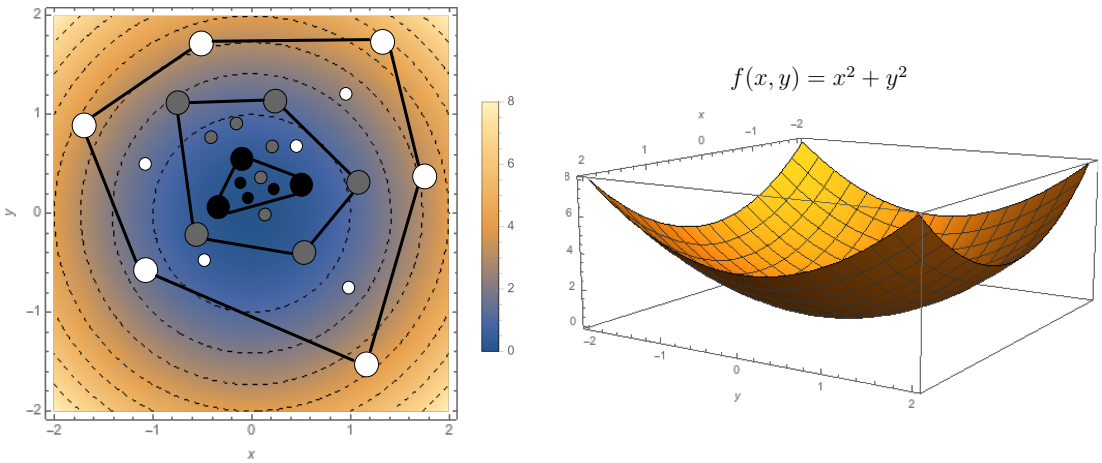
$\theta_i \geq 0$ and $\sum_{i=1}^{k} \theta_i = 1$. Note that the convex hull $co(A)$ is the set of all convex combinations $a' = \sum_{i=1}^{k} \theta_i a_i$, hence for each $a' \in co(A)$: $f(a') \leq \sum_{i=1}^{k} \theta_i f(a_i)$ [12].

Remark 9.3 (Definition 9.2)**.** All linear functions in vector spaces, including constant functions, are both convex and concave, thus affine [12].

To understand what Definition 9.2 means in terms of fitness and geometric crossovers, observe based on Figure 9.3 that:

- $\theta x + (1 - \theta)y$ is a geodesic interval on the real line representing offspring of parents x and y for some geometric crossover under Euclidean metric d_2; and,
- $\theta f(x) + (1 - \theta) f(y)$ is a weighted sum of the parents' fitnesses for each weight $\theta \in [0, 1]$. (Their arithmetic mean is obtained exactly when $\theta = \frac{1}{2}$.)

Therefore, for a traditional convex function and fixed parents and offspring under geometric crossover, Equation 9.1 indicates that the offspring fitness cannot be greater than a weighted sum of the parents' fitnesses. The same is true for offspring in the convex hull of parents, taking into account Remark 9.2 and that geometric crossovers produce offspring (i.e. convex combinations) within the convex hull of parents (see Section 6.2). For a random variable offspring, an analogous form [12] of Equation 9.1 describes the expected offspring fitness.



(a) **(b)**

Figure 9.4. Plot of the convex function $f(x, y) = x^2 + y^2$ in three-dimensional Euclidean metric space $(\mathbb{R}^3, d_2)$ in Figure 9.4b. Example of abstract convex evolutionary search, during three population generations, displayed over the contour plot of f in Figure 9.4a. *Legend*: first population ('white dots'), second population ('grey dots'), third population ('black dots'), boundaries of populations' convex hulls ('black thick lines'), contours of f ('dashed concentric circles'), and $f(x, y)$ values ('gradient colour').

Observe that the convex functions in Figures 9.3–9.4 have two salient characteristics:

- The high correlation between the function evaluation at a point and its distance from the global minima: the closer they are, the lower the function value.
- The convexity of their *lower contour sets* or *sub-level sets*, these being defined as $L_{\leq \ell}(f) \stackrel{\text{def}}{=} \{x \in X \mid f(x) \leq \ell\}$ for some arbitrary $f : X \to \mathbb{R}$ and $\ell \in \mathbb{R}$. Particularly, Figure 9.4a shows for $f(x, y) = x^2 + y^2$ that each *contour* where f is constant $f(x, y) = \ell$, see the 'dashed concentric circles', circumscribes a corresponding sub-level set $L_{\leq \ell}(f)$ that is convex: the line segment between any two points in a given 'circle' lies within such 'circle'.

The previous features of convex functions align well with abstract convex evolutionary search in Figure 9.4a: populations approach the global minimum while the mutual distance between individuals within each population progressively decreases through generations, and the nested chain of populations' convex hulls forms a nested chain of convex sets.

Indeed, geometric-crossover EAs produce offspring populations which improve over parent populations on average, whether in Euclidean or more general metric spaces, according to Proposition 9.1 below. But traditional convex functions must be generalised appropriately, and there are numerous ways to do it [40, 79, 90, 132]. One way is to reformulate Jensen's inequality (Equation 9.1) in terms of metrics as Equation 9.2 does to define *d-metric convex functions* and *convex fitness landscapes*.

Definition 9.3 (***d*-metric convex function** [133])**.** Let $(X, \mathcal{C})$ be the geodesic convex space of an arbitrary metric space (X, d). Let $f : C \to \mathbb{R}$ be a real-valued function defined on a geodesic convex subset $C \in \mathcal{C}$. Then, f is *d-metric convex* if for all $x, y \in C$, such that $d(x, y) \neq 0$, and all $z \in [x, y]_d$:

$$f(z) \leq \frac{d(y, z)}{d(x, y)} f(x) + \frac{d(x, z)}{d(x, y)} f(y) \ . \tag{9.2}$$

The triplet (X, d, f) is a *convex fitness landscape*.

Remark 9.4 (Definition 9.3)**.** *Strictly convex*, *concave* and *strictly concave* fitness landscapes are defined by replacing the relation $\leq$ in Equation 9.2 with relations: $<$, $\geq$, and $>$ respectively. *Affine* fitness landscapes are convex and concave. In normed vector spaces with metric $d(x, y) \stackrel{\text{def}}{=} \|x - y\|_2$, namely Euclidean distance, d-metric convex functions are convex in the traditional sense of Definition 9.2 [133].

Proposition 9.1 (**Convex landscapes: expected offspring fitness** [101])**.** Let (X, d, f) be any convex fitness landscape. Let a finite offspring population $P_{t+1} \subseteq X$ result from a finite parent population $P_t \subseteq X$, at generation $t \geq 0$, by applying a geometric crossover under metric d to pairs of parents, each of them sampled (with replacement[1]) uniformly at random from P_t. Then, the expected fitness $\mathbb{E}[f(z)]$ of any offspring individual $z \in P_{t+1}$ is less or equal than the mean fitness of P_t.

Proposition 9.1 (see Corollary 1 in [101]) relies on two assumptions:

- The probability of producing z from any given parents remains unchanged if the roles of first and second parent are exchanged: $\Pr\{Z = z \mid z \in \xi(x, y)\} = \Pr\{Z = z \mid z \in \xi(y, x)\}$, where ξ is a geometric crossover (so $z \in [x, y]_d$) and Z is a random variable representing all possible offspring of parents x, y.
- In an arbitrary metric segment $[x, y]_d$, the number of points at a certain distance t from the endpoint x equals those at distance t from the endpoint y. That is, $|\{z : z \in [x, y]_d \wedge d(x, z) = t\}| = |\{z : z \in [x, y]_d \wedge d(y, z) = t\}|$, for all $0 \leq t \leq d(x, y)$. Metric segments with such property will be called *regular*[2] (see Figure 9.5).

If these assumptions are not met, then the mean fitness of the parent population P_t could be biased[3] to some of its individuals more than others, in which case Proposition 9.1 may not hold: $\mathbb{E}[f(z)]$ may be greater than the mean fitness of P_t. Nevertheless, $\mathbb{E}[f(z)] \leq \max_{p \in P_t} f(p)$ would hold because for any $x, y \in P_t$ and $z \in P_{t+1}$ such that $z \in [x, y]_d : f(z) \leq \max\{f(x), f(y)\}$, provided that f is a d-metric convex function [101]. In other words, the expected offspring fitness would not be worse than the worst parent fitness.

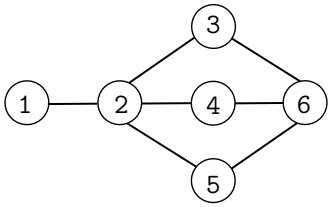


Figure 9.5. A graph where not all metric segments (i.e. shortest-paths) are regular. For example, consider points z in $[1, 6]_d$ at distance one from endpoint 1 and from endpoint 6 separately; d being the shortest-path metric. Then, $|\{z : z \in [1, 6]_d \wedge d(1, z) = 1\}| = |\{2\}| = 1 \neq 3 = |\{3, 4, 5\}| = |\{z : z \in [1, 6]_d \wedge d(z, 6) = 1\}|$.

[1] The probability of sampling the same element from a distribution is equal each time it is sampled.
[2] Regular metric segments are what Moraglio calls 'symmetric metric segments' [101]. Because any metric segment is symmetric as in $[x, y]_d = [y, x]_d$ [146], 'regular' is preferred to avoid confusion.
[3] This bias was noted also by Droste and Weismann [38] and recommended 'Guideline R 2' to prevent it.

Although Proposition 9.1 does not necessarily imply convergence to global optima nor efficiency of geometric-crossover EAs on convex fitness landscapes, it guarantees that, on average, they produce improving offspring populations or do not get worse than parents in the worst case. In this sense, GF justifies geometric-crossover EAs and convex fitness landscapes are well matched. Proposition 9.1 is true for any representation of solutions and any problem (regardless of dimension, instance, etc.), as long as its fitness function is d-metric convex and the aforesaid assumptions are accepted. Naturally, the guarantee can degrade when selection population operators give preference to individuals with worse fitness over individuals with better fitness.

9.3 Other General Convex Fitness Landscapes

Convex fitness landscapes (Definition 9.3) are a promising landscape class since they can be defined on general metric spaces, whether continuous or discrete, and are well matched with geometric-crossover EAs in the sense of Proposition 9.1. However, characterising which combinatorial or real-world problems belong or do not belong in this convex landscape class remains an open question in the GF. Chapter 11 will show in fact that convex fitness landscapes are not characteristic for certain classes combinatorial optimisation problems, as they become flat on a large class of finite graphs. Nevertheless, the following Sections 9.3.1–9.3.2 explain how the notion of d-metric convex function (Definition 9.3) can be 'approximated' or 'relaxed' to cover a larger class of landscapes other than such degenerate cases like flat landscapes.

9.3.1 Approximately Convex Fitness Landscapes

Moraglio's [101] idea of an *approximately convex fitness landscape* (Definition 9.4) is one whose fitness function is d-metric convex (Definition 9.3) if a bounded perturbation is tolerated, so that increasing or decreasing the tolerance increases or decreases the extent to which a fitness function is d-metric convex. Thus, approximately convex fitness landscapes are landscapes with a tunable degree of convexity. This generalises Hyers and Ulam's approximately convex functions [67] from Euclidean spaces to general metric spaces based on Soltan's d-metric convex functions [133].

Definition 9.4 (**d-metric ε-convex function** [67, 101])**.** Let $(X, \mathcal{C})$ be the geodesic convex space of an arbitrary metric space (X, d). Let $\varepsilon \in [0, \infty)$ and a real-valued function $f : C \to \mathbb{R}$ defined on a geodesic convex subset $C \in \mathcal{C}$. Then, f is *d-metric ε-convex*, or simply *(d, ε)-convex*, if for all $x, y \in C$, such that $d(x, y) \neq 0$, and all $z \in [x, y]_d$:

$$f(z) \leq \frac{d(y,z)}{d(x,y)} f(x) + \frac{d(x,z)}{d(x,y)} f(y) + \varepsilon \ . \tag{9.3}$$

The triplet (X, d, f) is an *approximately convex fitness landscape* with tolerance ε.

Remark 9.5 (Definition 9.4)**.** Approximately convex fitness landscapes (X, d, f) with a $(d, 0)$-convex fitness function f are convex fitness landscapes. *Strictly convex*, *concave* and *strictly concave* versions of approximately convex fitness landscapes are defined by: (a) replacing the relation $\leq$ in Equation 9.3 with relations $<$, $\geq$, and $>$ respectively; and, (b) replacing ε with $-\varepsilon$ for the concave versions.

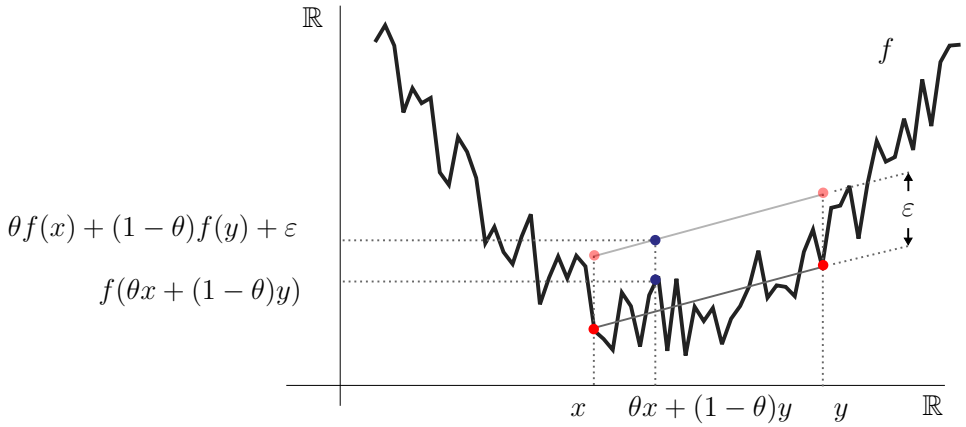


Figure 9.6. Visualisation of a (d_2, ε)-convex function, for some $\varepsilon > 0$, using Jensen's inequality in a two-dimensional Euclidean metric space $(\mathbb{R}^2, d_2)$.

Thus approximately convex fitness landscapes extend convex fitness landscapes because (d, ε)-convex functions (Equation 9.3) extend d-metric convex functions (Equation 9.1): the former add a tolerance ε to the latter to allow varying degrees of perturbation in the fitness function. If this tolerance is large enough, then effectively any fitness landscape would be approximately convex according to Definition 9.4. However, this does not mean any actual fitness landscape of combinatorial or real world problems necessarily has a 'globally convex' trend resembling, for example, the one illustrated in Figure 9.6. Notice that increasing the tolerance allows higher and higher degrees of perturbation in the fitness function, which can break globally convex trends. For instance, according to Definition 9.4, degenerate cases like a fitness function with fitness values randomly chosen within $[0, 1]$ would also be approximately convex if the tolerance chosen is $\varepsilon = 1$. The relevance of approximately convex fitness landscapes very much depends on whether the fitness functions of interesting optimisation problems are (d, ε)-convex for relatively small ε.

Earlier in Section 9.2, Proposition 9.1 justified that geometric-crossover EAs are well matched with convex fitness landscapes because of the fitness improvement expected for any offspring population over its parent population. Proposition 9.2 (see Theorem 5 in [101]) proves the analogous result of Proposition 9.1 for approximately convex fitness landscapes.

Proposition 9.2 (**Approx. convex landscapes: expected offspring fitness** [101])**.** Let (X, d, f) be any approximately convex fitness landscape with a (d, ε)-convex fitness function f. Let a finite offspring population $P_{t+1} \subseteq X$ result from a finite parent population $P_t \subseteq X$, at generation $t \geq 0$, by applying a geometric crossover under metric d to pairs of parents, each of them sampled (with replacement) uniformly at random from P_t. Then, the expected fitness $\mathbb{E}[f(z)]$ of any offspring individual $z \in P_{t+1}$ fulfils: $\mathbb{E}[f(z)] \leq \overline{P_t} + \varepsilon$, where $\overline{P_t}$ is the mean fitness of P_t.

The potential impact the tolerance ε has on geometric-crossover EAs performance is noticeable due to Proposition 9.2. Increasing ε allows fitness landscapes to be less convex, thereby diminishing the likelihood that a geometric-crossover EA generates, on average, offspring populations improving over their parent populations. Notice $\mathbb{E}[f(z)] \leq \overline{P_t} + \varepsilon$ does not necessarily imply $\mathbb{E}[f(z)] \leq \overline{P_t}$; when $\varepsilon > 0$, it may be true that $\mathbb{E}[f(z)] > \overline{P_t}$ without falsifying Proposition 9.2. Consequently, a relatively small ε is important for keeping a globally convex trend in the landscape and avoid damaging geometric-crossover EAs performance.

9.3.2 Quasi-convex and Average-convex Fitness Landscapes

Quasi-convex and average-convex fitness landscapes are two alternative ways to generalise convex fitness landscapes (Section 9.2) without perturbing the fitness function like approximately convex fitness landscapes (Section 9.3.1). Instead, they 'relax' the notion of convexity in d-metric convex functions (Definition 9.3) while retaining distinctive properties thereof that make geometric-crossover EAs well matched with quasi-convex and average-convex fitness landscapes.

Quasi-convex Fitness Lansdcapes

Moraglio [101] proposes *quasi-convex fitness landscapes*, defined in terms of *d-metric quasi-convex functions*, based on Singer [132] and Soltan [133]. These functions have the characteristic that every sub-level set must be geodesically convex.

Definition 9.5 (*d*-**metric quasi-convex function** [101])**.** Let $(X, \mathcal{C})$ be the geodesic convex space of a metric space (X, d). Let a real-valued function $f : C \to \mathbb{R}$ defined on a geodesic convex subset $C \in \mathcal{C}$. Then, f is *d-metric quasi-convex* if every sub-level set $L_{\leq \ell}(f) \stackrel{\text{def}}{=} \{x \in X \mid f(x) \leq \ell\}$ is geodesically convex. That is, for all $\ell \in \mathbb{R}$ and $x, y \in L_{\leq \ell}(f)$: $[x, y]_d = \{z \in X \mid d(x, z) + d(z, y) = d(x, y)\} \subseteq L_{\leq \ell}(f)$. The triplet (X, d, f) is a *quasi-convex fitness landscape*.

Remark 9.6 (Definition 9.5)**.** *Quasi-concave fitness landscapes* are defined similarly via super-level sets $L_{\geq \ell}(f) \stackrel{\text{def}}{=} \{x \in X \mid f(x) \geq \ell\}$. *Quasi-linear fitness landscapes* are quasi-convex and quasi-concave.

Definition 9.5 of d-metric quasi-convex functions generalises Definition 9.3 of d-metric convex functions because: any d-metric convex function has geodesically convex sub-level sets according to Proposition 9.3 below (see Theorem 2 in [133]), but not all functions with geodesically convex sub-level sets are d-metric convex functions necessarily. In other words, convexity implies quasi-convexity, but the converse is false in general. A simple example of a quasi-convex function in Euclidean space that is not convex is $f(x) = \sqrt{|x|}$ plotted in Figure 9.7: the line segment, or chord, between two points $(x_1, f(x_1))$ and $(x_2, f(x_2))$ in the graph of f lies below the graph not above it as required by Jensen's inequality (Equation 9.1) of convex functions.

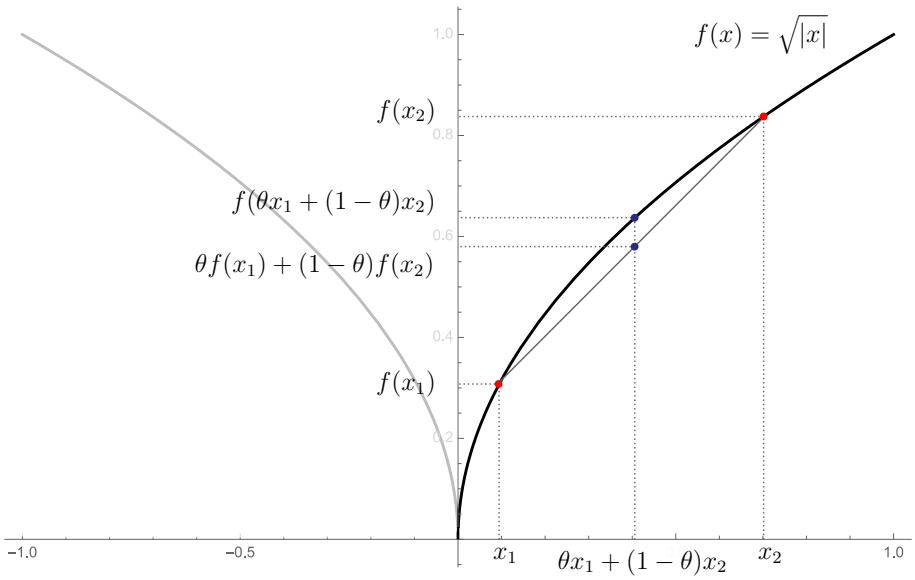


Figure 9.7. Visualisation of a d_2-metric quasi-convex function f in two-dimensional Euclidean metric space $(\mathbb{R}^2, d_2)$. The function does not fulfil Jensen's inequality $f(\theta x_1 + (1-\theta)x_2) \leq \theta f(x_1) + (1-\theta) f(x_2)$, so it is not convex, but it fulfils $f(\theta x_1 + (1-\theta)x_2) \leq \max\{f(x_1), f(x_2)\}$ where $\theta \in [0, 1]$.

Proposition 9.3 (*d*-**metric convex function: sub-level sets** [133])**.** Let $(X, \mathcal{C})$ be the geodesic convex space of an arbitrary metric space (X, d). If $f : C \to \mathbb{R}$ is a d-metric convex function on a geodesic convex subset $C \in \mathcal{C}$, then the *sub-level sets* $L_{\leq \ell}(f) \stackrel{\text{def}}{=} \{x \in X \mid f(x) \leq \ell\}$ are geodesically convex; that is, $\forall \ell \in \mathbb{R}$ $\forall x, y \in L_{\leq \ell}(f) : [x, y]_d \subseteq L_{\leq \ell}(f)$. The same is true of *strict sub-level sets* $L_{<\ell}(f) \stackrel{\text{def}}{=} \{x \in X \mid f(x) < \ell\}$.

A quasi-concave fitness landscape which is not convex nor concave is $(\{0,1\}^n, d_{\mathrm{H}}, f_{\textsc{LeadingOnes}})$ proven in Chapter 11: $(\{0,1\}^n, d_{\mathrm{H}})$ is a n-dimensional binary Hamming metric space, and $f_{\textsc{LeadingOnes}}(x) \stackrel{\text{def}}{=} \sum_{i=1}^{n} \prod_{j=1}^{i} x_j \ \forall x \in \{0,1\}^n$ is the 'leading-ones' function that counts the number of consecutive left-most bits set to one in x.

Although d-metric quasi-convex functions generally do not satisfy the original Jensen's inequality of d-metric convex functions, they always satisfy the following 'weaker' forms in Proposition 9.4 which derives easily[4] from Definition 9.5.

Proposition 9.4 (*d*-**metric quasi-convex functions: inequalities** [101, 104])**.** Let (X, d) be any metric space and $f : X \to \mathbb{R}$ a d-metric quasi-convex function. Then, the following two inequalities are true:

(a) $\forall x, y \in X$ and $\forall z \in [x, y]_d$: $f(z) \leq \max\{f(x), f(y)\}$; and,

(b) $\forall P \subseteq X$ and $\forall z \in co(P)$: $f(z) \leq \max_{p \in P} f(p)$.

Proposition 9.4 has relevant implications concerning geometric-crossover EAs performance on quasi-convex fitness landscapes comparable to those discussed earlier in Section 9.2 for convex fitness landscapes. Regardless of a specific representation of solutions and specific problem, Proposition 9.4 means that fitnesses of individuals in an offspring population P_{t+1} are never worse than the worst fitness (i.e. maximum fitness) of its parent population P_t at any generation t, considering that $P_{t+1} \subseteq co(P_{t+1}) \subseteq co(P_t)$ due to abstract convex evolutionary search. That is, for any offspring $z \in P_{t+1}$, its fitness $f(z) \leq \max_{p \in P_t} f(p)$. In this sense, GF regards geometric-crossover EAs and quasi-convex fitness landscapes well matched. Remarkably, for a restricted but still general subclass of quasi-convex landscapes, certain geometric-crossover EAs provably find their global optima in a polynomial number of generations and fitness evaluations [104].

[4] Let (X, d, f) be a quasi-convex landscape and any $x, y \in X$. Suppose $f(x) \leq f(y)$ without loss of generality. Then, $x, y \in L_{\leq f(y)}(f)$, so $[x, y]_d \subseteq L_{\leq f(y)}(f)$ because f is quasi-convex. Therefore, any $z \in [x, y]_d$ fulfils $f(z) \leq f(y)$ since $z \in L_{\leq f(y)}(f)$, hence $f(z) \leq \max\{f(x), f(y)\}$.

Average-convex Fitness Lansdcapes

Average-convex fitness landscapes are those where the expected fitness of any offspring uniformly distributed over the metric segment between its parents is not greater than the arithmetic mean fitness of such parents. In essence, average-convex fitness landscapes make Proposition 9.1 of convex fitness landscapes their own definition, without them being convex fitness landscapes necessarily.

Definition 9.6 (*d*-**metric average-convex function** [101, 104])**.** Let a real-valued function $f : X \to \mathbb{R}$ be defined on any metric space (X, d). Then, f is a *d-metric average-convex* if for all $x, y \in X$ and all $z \sim \text{Unif}([x, y]_d)$: $\mathbb{E}[f(z)] \leq \frac{1}{2}\Big(f(x)+f(y)\Big)$. The triplet (X, d, f) is an *average-convex fitness landscape*.

Remark 9.7 (Definition 9.6)**.** *Average-concave fitness landscapes* are defined analogously by replacing the relation $\leq$ in Definition 9.6 with the relation $\geq$. *Average-affine fitness landscapes* are average-convex and average-concave.

For convex fitness landscapes, the expected fitness in an offspring population is not worse than its parent population's mean fitness due to Proposition 9.1. Evidently, the same would be true for average-convex fitness landscapes given that Definition 9.6 of d-metric average-convex functions is a particular case of the above restricted to populations with only two parents. Therefore, geometric-crossover EAs would perform on average-convex fitness landscapes comparably as well as on convex fitness landscapes.

The principal advantage of average-convex fitness landscapes is being a less restrictive class than convex fitness landscapes. Consider $(\{0,1\}^n, d_{\text{H}}, f_{\textsc{OneMax}})$ where $(\{0,1\}^n, d_{\text{H}})$ is a n-dimensional binary Hamming metric space with Hamming metric d_{H}, and $f_{\textsc{OneMax}}(x) \stackrel{\text{def}}{=} \sum_{i=1}^{n} x_i$ is the 'one-max' function that counts the number of bits set to one of any given $x \in \{0,1\}^n$. Chapter 11 will prove $(\{0,1\}^n, d_{\text{H}}, f_{\textsc{OneMax}})$ is an average-affine fitness landscape, thus average-convex and average-concave, but neither convex nor concave.

Approximately Quasi-convex and Average-convex Fitness Landscapes

Convex fitness landscapes have been generalised adopting two separate approaches so far: either by approximating their definition, leading to approximately convex fitness landscapes, or by relaxing their definition, leading to quasi-convex and average-convex fitness landscapes. These can be further generalised to *approximately quasi-convex* and *approximately average-convex* fitness landscapes, combining both approaches as in the following Definitions 9.7–9.8.

Definition 9.7 (*d*-**metric** ε-**quasi-convex function** [101])**.** Let $(X, \mathcal{C})$ be the geodesic convex space of an arbitrary metric space (X, d). Let $\varepsilon \in [0, \infty)$ and a real-valued function $f : C \to \mathbb{R}$ defined on a geodesic convex subset $C \in \mathcal{C}$. Then, f is *d-metric ε-quasi-convex*, or simply (d, ε)-*quasi-convex*, if for all $x, y \in C$ and all $z \in [x, y]_d : f(z) \leq \max\{f(x), f(y)\} + \varepsilon$. The triplet (X, d, f) is an *approximately quasi-convex fitness landscape* with tolerance ε.

Definition 9.8 (*d*-**metric** ε-**average-convex function** [104])**.** Let $\varepsilon \in [0, \infty)$ and real-valued function $f : X \to \mathbb{R}$ defined on an arbitrary metric space (X, d). Then, f is *d-metric ε-average-convex*, or simply (d, ε)-*average-convex*, if for all $x, y \in X$ and all $z \sim \text{Unif}([x, y]_d) : \mathbb{E}[f(z)] \leq \frac{1}{2}\Big(f(x) + f(y)\Big) + \varepsilon$. The triplet (X, d, f) is an *approximately average-convex fitness landscape* with tolerance ε.

Remark 9.8 (To Definitions 9.7–9.8)**.** *Approximately quasi-concave* and *approximately average-concave fitness landscapes* are defined analogously by replacing the inequalities $\leq$ with $\geq$ and the tolerances ε with $-\varepsilon$.

Regarding geometric-crossover EAs performance, no formal guarantees analogous to Proposition 9.2 on the expected offspring fitness in approximately convex fitness landscapes have been proved yet for approximately quasi-convex and average-convex fitness landscapes [101, 104]. Nevertheless, from Section 9.3.1 and Section 9.3.2 earlier, one may argue that the expected fitness in offspring populations would not be worse than the average or worst fitness of their parent populations; provided that perturbations in the fitness function are bounded by a sufficiently small tolerance ε, thus preserving a 'global convex' trend to some degree. Obviously, for $\varepsilon = 0$ the approximately quasi-convex and average-convex classes reduce to their non-approximated quasi-convex and average-convex counterparts. The question here is whether the fitness functions of relevant optimisation problems actually fall into the above classes for a small non-zero ε.

Chapter 10

Fitness Landscape Decomposition in Elementary Landscapes Theory

This chapter revises background material for subsequent chapters. It summarises and clarifies key ideas in ELT to analyse general fitness functions defined on finite graphs and recombination P-structures introduced in Chapter 4. This chapter principally supports the contributions in Chapter 11 analysing the convexity of fitness functions for certain well known combinatorial problems.

10.1 Fitness Landscape Decomposition

A primary aim of ELT is analysing fitness landscapes associated with combinatorial optimisation problems to understand the efficiency and efficacy of EAs searching in those landscapes [5, 35, 136–138, 140, 150]. ELT proposes a fitness landscape analysis that basically consists in decomposing fitness landscapes into simpler and separate 'components', relying on fundamental notions in linear algebra and spectral graph theory [8, 13, 25, 113]. Such analysis facilitates a rigorous understanding of the relationship between the fitness functions of combinatorial problems, the search operators of EAs driving evolutionary change, namely mutation and recombination, and the underlying search space structure associated with them. This chapter explains how ELT formalises fitness landscape decomposition. First, Section 10.1.1 introduces some necessary preliminary notions based on the above bibliography.

10.1.1 Basics of Fitness Landscape Decomposition

To decompose fitness functions of combinatorial optimisation problems, whether defined on mutation or recombination search spaces (Chapter 4), ELT considers the

set of all real-valued fitness functions $\mathbb{R}^X = \{f : X \to \mathbb{R}\}$ with a finite domain set $X = \{x_0, x_1, \ldots, x_{n-1}\}$, $|X| = n$. The set $\mathbb{R}^X$ forms an *inner product vector space* where the *inner product* is the scalar product

$$\langle f, g \rangle \stackrel{\text{def}}{=} \sum_{x \in X} f(x)g(x) \ , \quad \forall f, g \in \mathbb{R}^X \ . \tag{10.1}$$

Elements in $\mathbb{R}^X$ are treated as functions or vectors. Moreover, if $\{\varphi_0, \varphi_1, \ldots, \varphi_{n-1}\}$ is an *orthonormal basis* of functions that spans $\mathbb{R}^X$, then every fitness function $f \in \mathbb{R}^X$ is uniquely described as a *linear combination*

$$f(x) = \sum_{i=0}^{n-1} a_i \varphi_i(x) \tag{10.2}$$

of the basis functions φ_i and the scalars $a_i = \langle \varphi_i, f \rangle$ that represent the magnitude of f projected onto φ_i. Equation 10.2 can be written $f = a_0\varphi_0 + \cdots + a_{n-1}\varphi_{n-1}$ in vectorial form, from which it is clear there is a one-to-one correspondence between a function $f \in \mathbb{R}^X$ and a real vector $(a_0, a_1, \ldots, a_{n-1}) \in \mathbb{R}^n$ once $\{\varphi_i\}$ is fixed. That is, $\mathbb{R}^X$ is isomorphic to $\mathbb{R}^{|X|} = \mathbb{R}^n$.

Figure 10.1a intuitively illustrates what a decomposition would look like for an hypothetical real-valued function f defined on $X = \{x_0, x_1, x_2\}$, where $f(x_i) = a_i$ for some $a_i \in \mathbb{R}$, by assigning a coordinate system to X based on the standard basis functions in $\mathbb{R}^3$: $\varphi_0 = (1, 0, 0)$, $\varphi_1 = (0, 1, 0)$, $\varphi_2 = (0, 0, 1)$. Here f decomposes into three functions: $a_0\varphi_0 = (a_0, 0, 0)$, $a_1\varphi_1 = (0, a_1, 0)$ and $a_2\varphi_2 = (0, 0, a_2)$.

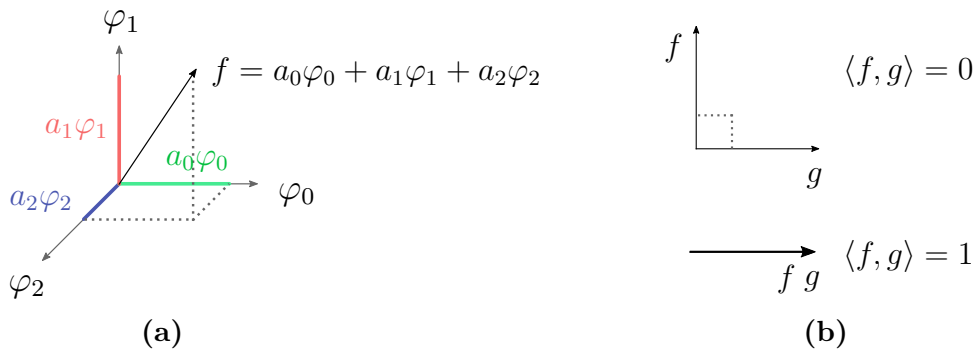


Figure 10.1. A linear combination f of orthonormal basis functions φ_i and scalars a_i in Figure 10.1a, and the orthogonality of two normalised functions f and g with respect to their inner product in Figure 10.1b.

Equation 10.2 provides the means to decompose combinatorial fitness functions, but it is not clear how it relates to the underlying structure of mutation or recombination search spaces where ELT defines fitness functions. The next Sections 10.2–10.3

explain how that is accomplished by choosing the orthonormal basis of appropriate symmetric linear operators. A linear operator $\mathbf{M} : \mathbb{R}^X \to \mathbb{R}^X$ is *symmetric* or *self-adjoint* if and only if $\langle \mathbf{M}u, v \rangle = \langle u, \mathbf{M}v \rangle$ for all $u, v \in \mathbb{R}^X$. Juxtaposition $\mathbf{M}u$ means $\mathbf{M}$ applied to u, so $\mathbf{M}u \in \mathbb{R}^X$, or ordinary matrix multiplication viewing $\mathbf{M}$ as a matrix and u as a column vector.

10.2 Mutation Fitness Landscapes

A *mutation fitness landscape* (X, N, f) consists of a fitness function $f : X \to \mathbb{R}$ defined on a mutation search space (X, N), where X is a finite set of all candidate solutions, and $N : X \to \mathcal{P}(X)$ is a neighbourhood function that describes the support sets of a mutation operator by mapping $x \in X$ to a subset of neighbouring solutions $N(x) \in \mathcal{P}(X)$ [136]. Mutation search spaces are associated with finite connected graphs (Chapter 4). Sometimes (X, N, f) will be more conveniently denoted (G, f) and leave the neighbourhood implicit in the graph $G = (V, E)$ with vertex set $V = X$ and edge set $E = \{(x, y) \mid y \in N(x),\ x, y \in V\}$. Unless stated otherwise, N is symmetric, thus G is undirected. Extensions for directed and irregular graphs [5, 35] are not addressed in this thesis since most of ELT develops upon connected, simple, undirected and regular graphs.

10.2.1 Decomposition via Discrete Laplacian Operators

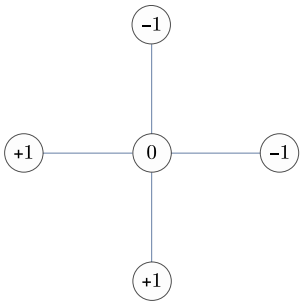
To decompose mutation fitness landscapes (X, N, f) via Equation 10.2, ELT chooses an orthonormal basis of a *symmetric linear operator* called *discrete Laplacian operator* [8, 136, 137] relating the fitness f and neighbourhood N functions

$$\begin{aligned} &\mathbf{L} : \mathbb{R}^X \to \mathbb{R}^X \ , \\ &(\mathbf{L}f)(x) \stackrel{\text{def}}{=} \sum_{y \in N(x)} \Big(f(x) - f(y) \Big) = |N(x)|\, f(x) - \sum_{y \in N(x)} f(y) \ . \end{aligned} \tag{10.3}$$

Plainly, $\mathbf{L}$ transforms an input fitness function f to a function $\mathbf{L}f$ that sums the differences in fitness between a given individual x and its neighbours $y \in N(x)$. These finite differences are reminiscent of 'discrete derivatives' for graphs. In spectral graph theory [13], it is customary to write $\mathbf{L}$ (Equation 10.3) as the *graph Laplacian matrix* (of dimensions $|X| \times |X|$) for the graph G associated with N

$$\mathbf{L} \stackrel{\text{def}}{=} \mathbf{D} - \mathbf{A} \ , \tag{10.4}$$

D and **A** being the vertex degree and adjacency matrices of G. An orthonormal basis $\{\varphi_i\}$ of $\mathbf{L}$ consists of the *eigenfunctions* or *eigenvectors* φ_i of $\mathbf{L}$, hence $(\mathbf{L}\varphi_i)(x) = \lambda_i\varphi_i(x)$ and λ_i is the *eigenvalue* corresponding to φ_i. Note $\mathbf{L}f$ is a linearly transformed function, whereas $(\mathbf{L}f)(x)$ is the evaluation of such transformed function at x. Figure 10.2 shows a Laplacian eigenfunction for the star graph S_5.



$$\begin{matrix} & & & \mathbf{L} & & & f & & \mathbf{L}f \end{matrix}$$

$$\begin{bmatrix} 4 & -1 & -1 & -1 & -1 \\ -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}$$

Figure 10.2. A function $f : V \to \{-1, 0, +1\}$ defined on the vertex set V of a star graph S_5 ('left') that is an eigenfunction of the graph Laplacian matrix $\mathbf{L}$ of S_5 with eigenvalue one since the column vectors f and $\mathbf{L}f$ are equal ('right').

Then, using Equation 10.2, every fitness function $f \in \mathbb{R}^X$ in a mutation fitness landscape (X, N, f) can be decomposed according to an orthonormal basis of eigenfunctions for the graph Laplacian associated with the neighbourhood function N (Equations 10.3–10.4). Such decomposition is called a *Fourier series expansion* of f for its analogy with Fourier series analysis of periodic functions like signals [136, 137]. Hereinafter, the eigenvalues of $\mathbf{L}$ are enumerated in increasing order and starting at index zero

$$\lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{i-1} < \underbrace{\lambda_i = \lambda_{i+1} = \cdots = \lambda_{i+m_i-1}}_{m_i \text{ times}} < \lambda_{i+m_i} \leq \cdots \leq \lambda_{|X|-1} \ , \quad (10.5)$$

where m_i is the *multiplicity* of the eigenvalue λ_i. This enumeration is called the *spectrum*[1,2] of $\mathbf{L}$ denoted as a multiset of the form $Spectrum(G) = \{\lambda_{i_{[m_i]}}\}$, for a given graph G, and the subscript $[m_i]$ indicates the number of occurrences of each λ_i. The *order* p of an eigenvalue is its index or position in the spectrum without counting multiplicities [8]. For instance, the star graph S_5 in Figure 10.2 has $Spectrum(S_5) = \{0_{[1]}, 1_{[3]}, 5_{[1]}\} = \{0, 1, 1, 1, 5\}$ [13], so the example eigenfunction f corresponds to the eigenvalue $\lambda_p = 1$ of order $p = 1$.

[1]The eigenvalue $\lambda_0 = 0$ corresponds to the constant (all-ones) eigenvector $\varphi_0 = \mathbf{1} = (1, 1, \ldots, 1)$ and, for connected graphs, $\lambda_0 = 0$ has multiplicity one [136].

[2]In spectral graph theory, the spectrum of a graph may also refer to the eigenvalues of the adjacency matrix [13]. This thesis always refers to the spectrum of the graph Laplacian.

Example: Pseudo-Boolean Fitness Functions in the Walsh Basis

A class of fitness functions well known in combinatorial optimisation are *pseudo-Boolean functions*.

Definition 10.1 ([k-bounded] **pseudo-Boolean function** [11])**.** Let $J \subseteq \{1, \ldots, n\}$ be an index set for any $n \in \mathbb{N}$. A *pseudo-Boolean function* is any function $f : \{0,1\}^n \to \mathbb{R}$ uniquely represented as a multi-linear polynomial

$$f(x) \stackrel{\text{def}}{=} \sum_{S \subseteq J} c_S \left(\prod_{j \in S} x_j \right) \ , \tag{10.6}$$

where c_S are real number coefficients and $x = (x_1, \ldots, x_n) \in \{0,1\}^n$. By convention, $\prod_{j \in \emptyset} x_j = 1$. If the cardinality of the largest subset S for $c_S \neq 0$ is $|S| \leq k$, then f is *k-bounded*; that is, the bit-wise non-linearity of f is at most k bits.

Example 10.1 (**2-bounded pseudo-Boolean function**)**.** A simple 2-bounded pseudo-Boolean function is $f(x) = -\frac{1}{2}x_1 + x_1x_2$, for an index set $J = \{1,2\}$ and coefficients: $c_\emptyset = 0$, $c_{\{1\}} = -\frac{1}{2}$, $c_{\{1,2\}} = 1$. The function values are: $f(\texttt{00}) = 0$, $f(\texttt{01}) = 0$, $f(\texttt{10}) = -\frac{1}{2}$ and $f(\texttt{11}) = \frac{1}{2}$.

The space of all pseudo-Boolean functions $\mathbb{R}^{\{0,1\}^n} = \{f : \{0,1\}^n \to \mathbb{R}\}$, isomorphic to $\mathbb{R}^{2^n}$, is spanned by an orthogonal basis that has been studied extensively in relation with GAs and Holland's schemas: the *Walsh basis* [54, 55, 65, 124, 125, 151]. In other words, any pseudo-Boolean fitness function linearly decomposes into a sum of *Walsh basis functions*[3] ψ_w indexed by each $w \in \{0,1\}^n$: $\psi_w(x) \stackrel{\text{def}}{=} \prod_{i=1}^n (-1)^{w_i x_i}$ for all $x \in \{0,1\}^n$. For notational convenience, the (non-normalised) Walsh functions ψ_w can be written using the equivalent alphabet $\{-1,+1\}$, rather than the usual $\{0,1\}$, for each index set $J \subseteq \{1, \ldots, n\}$ [8, 85, 137]:

$$\psi_J(x) \stackrel{\text{def}}{=} \prod_{j \in J} x_j \ , \quad \forall x \in \{-1,+1\}^n \ . \tag{10.7}$$

Each ψ_J is an eigenfunction of the graph Laplacian $\mathbf{L}$ for binary Hamming graphs induced by the single bit-flip neighbourhood seen in Chapter 4:

$$\mathbf{L}\psi_J = 2p \ \psi_J \ , \tag{10.8}$$

with eigenvalue $\lambda_p = 2p$ of order $p = |J|$ and multiplicity $\binom{n}{|J|}$ [8, 85, 137]. Figure 10.3 below shows the $2^3 = 8$ possible Walsh functions on a three-dimensional bi-

[3] Walsh functions are also called *p-spin* functions (as in the spin-glass model from physics) [139, 150].

nary Hamming graph. For example, consider the order-two Walsh functions $\psi_{\{1,3\}} = x_1x_3$ and $\psi_{\{2,3\}} = x_2x_3$ in Figure 10.3 and the binary sequence $x = (+1, -1, +1)$ corresponding to the vertex label +−+. Then, $\psi_{\{1,3\}}(x) = (+1) \cdot (+1) = +1$ shown as a 'red vertex', and $\psi_{\{2,3\}}(x) = (-1) \cdot (+1) = -1$ shown as a 'blue vertex'. Klemm, Stadler [85], and Goldberg [54] show alternative visualisations of these Walsh functions.

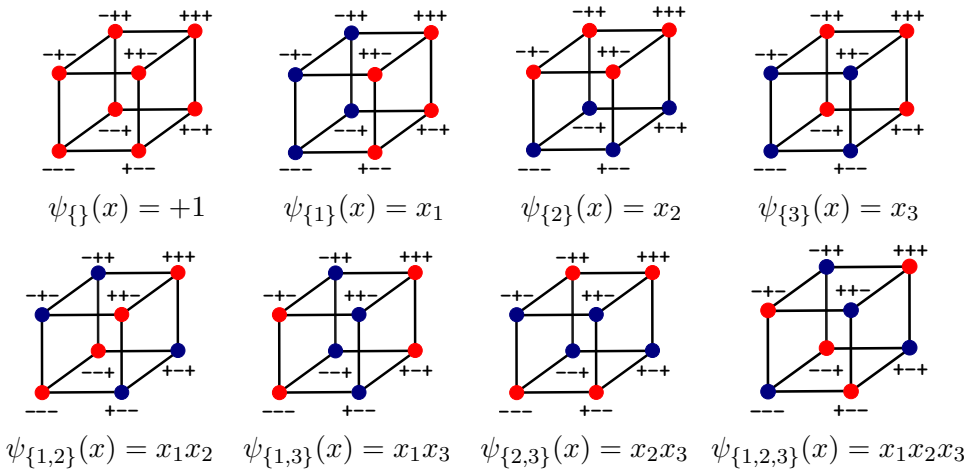


Figure 10.3. The eight possible Walsh basis functions $\psi_J(x)$ for $J \subseteq \{1, 2, 3\}$ and $x \in \{-1, +1\}^3$ on a three-dimensional binary Hamming graph. Vertices correspond to three-dimensional binary Hamming sequences using the alphabet $\{-1, +1\}$. A 'red' vertex indicates that ψ_J evaluates to $+1$ and a 'blue' vertex that ψ_J evaluates to -1.

10.2.2 Elementary Fitness Landscapes

The Fourier series expansion of a fitness function presented in Section 10.2.1 puts at our disposal the powerful machinery of spectral graph theory to analyse fitness functions defined on graphs (e.g. their connectivity, fitness correlation, estimated number of local optima, etc.) [8, 13, 136, 137]. But would such series expansion be practical actually? Although the series in Equation 10.2 always expands into finitely many terms because the solution set X is assumed finite, in general there could be an exponentially large number of terms to evaluate (e.g. up to $|\{0, 1\}^n| = 2^n$ for pseudo-Boolean functions), thereby undermining the practicality of fitness function decomposition.

However, for many NP-complete problems in combinatorial optimisation, the Fourier series expansion of their fitness function has only a few non-zero terms with a localised spectrum, meaning their Fourier series expansion does not comprise an exponentially large number of terms [85, 136, 137]. For instance, the fitness func-

tion of the subset sum problem and the quadratic assignment problem decompose into a sum of two and, respectively, three Laplacian eigenfunctions [20]. Indeed, Grover [56] discovered that the fitness function of several NP-complete problems is itself a Laplacian eigenfunction up to an additive constant[4] c for common choices of neighbourhood functions. So that the Fourier series expansion (Equation 10.2) reduces to an even simpler form

$$f(x) = c + \varphi(x) \ ,$$

where φ is an eigenfunction of the graph Laplacian. Such class of fitness landscapes constitute the foundations of ELT hence their name *elementary landscapes* (ELs) as in Definition 10.2. Any elementary landscape (EL) has, therefore, a graph Laplacian eigenfunction as fitness function. Key characteristics of ELs regarding their local optima and global structure are discussed later in Section 10.4.

Definition 10.2 ([**Flat/non-flat**] **elementary landscape** [136, 137])**.** Let a fitness landscape $\mathscr{F}$ be defined by a fitness function $f : V \to \mathbb{R}$ on a finite connected graph G. $\mathscr{F}$ is *elementary* if and only if f is a graph Laplacian eigenfunction φ_p with eigenvalue $\lambda_p \geq 0$ of order $p \geq 0$, so that: $(\mathbf{L}f)(v) = \lambda_p f(v)$, $\forall v \in V$. If f is φ_0 with $\lambda_0 = 0$ and $p = 0$, then $\mathscr{F}$ is *flat elementary* because φ_0 is the constant all-ones eigenvector $\mathbf{1}$. If f is φ_p with $\lambda_p > 0$ and $p > 0$, then $\mathscr{F}$ is *non-flat elementary* because $\langle \varphi_p, \varphi_0 \rangle = 0$: φ_p is orthogonal to $\mathbf{1}$, so it is non-constant.

Remark 10.1 (Definition 10.2)**.** A fitness function f offset by a scalar $c \in \mathbb{R}$ as in $f' = f - c\mathbf{1}$ may be a graph Laplacian eigenfunction even if f is not. Offsetting all fitness values by an equal amount does not fundamentally change the fitness function, so a fitness landscape (G, f) can be called *elementary* if and only if (G, f') is elementary for at least one scalar offset c (see Corollary 4 in [5]).

From Definition 10.2, it is clear that the Walsh functions shown in Figure 10.3 earlier are examples of ELs on a three-dimensional binary Hamming graph since they are eigenfunctions of the associated graph Laplacian (Equation 10.8). Chapter 11 will analyse and give examples of simple and NP-complete problems in combinatorial optimisation whose fitness function corresponds to a Walsh function.

Remark 10.2 (Definition 10.2)**.** Each eigenfunction corresponds to an eigenvalue at a specific position in the spectrum. If two ELs are the same, their fitness function

[4]The additive constant c is the global average $\bar{f}$ of f if $\langle \varphi, \varphi_0 \rangle = \langle \varphi, \mathbf{1} \rangle = 0$ [137]. Notice: $|X|\,\bar{f} = |X|\,\frac{1}{|X|}\sum_{x \in X} f(x) = \sum_{x \in X}(c + \varphi(x)) = |X|\,c + \sum_{x \in X}\varphi(x) = |X|\,c + \langle \varphi, \mathbf{1} \rangle = |X|\,c$.

is the same graph Laplacian eigenfunction at the same position and thus of the same order p. However, two eigenfunctions of equal order need not be the same eigenfunction when the eigenvalue's multiplicity is not one (Figure 10.4).

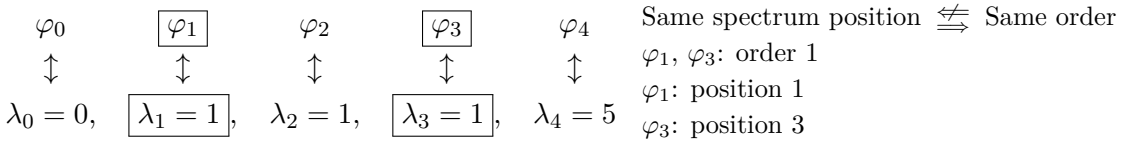


Figure 10.4. Graph Laplacian eigenfunctions of same order need not be the same eigenfunction. This figure shows the relationship between the order and position of graph Laplacian eigenfunctions $\varphi_0, \ldots, \varphi_4$ corresponding to eigenvalues in the spectrum $Spectrum(S_5) = \{0_{[1]}, 1_{[3]}, 5_{[1]}\} = \{0, 1, 1, 1, 5\}$ of the star graph S_5 in Figure 10.2.

Elementary Landscapes via Random Walks

The transition probability matrix $\mathbf{P} \stackrel{\text{def}}{=} \mathbf{A}\mathbf{D}^{-1}$ of a random walk on a graph (see Equation 7.1) provides an alternative way to define ELs as revealed by the identity

$$\mathbf{D}^{-1}\mathbf{L} = \mathbf{D}^{-1}(\mathbf{D} - \mathbf{A}) = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A} = \mathbf{I} - \mathbf{P}^{\mathrm{T}} \ , \tag{10.9}$$

relating the transpose $\mathbf{P}^{\mathrm{T}}$ of $\mathbf{P}$ and a normalisation $\mathbf{D}^{-1}\mathbf{L}$ of the graph Laplacian $\mathbf{L}$, $\mathbf{I}$ being the identity matrix [8]. For D-regular graphs, Equation 10.9 simplifies to $\mathbf{P} = \mathbf{I} - \frac{1}{D}\mathbf{L}$, where $\mathbf{P}$ and $\mathbf{L}$ both are symmetric linear operators with an identical orthogonal basis of eigenfunctions, that is identical ELs (see Lemma 12 in [136]). (Besides $\mathbf{P} = \mathbf{P}^{\mathrm{T}}$, the symmetry of $\mathbf{P}$ is justified if one recalls from Chapter 7 that $\mathbf{P}$ induces a reversible finite Markov chain.) Therefore, a mutation fitness landscape is called *elementary* equivalently when its fitness function is an eigenfunction of the transition probability matrix of a random walk on that landscape [136, 137].

10.3 Recombination Fitness Landscapes

A *recombination fitness landscape* $(X, \mathcal{R}, f)$ consists of a fitness function $f : X \to \mathbb{R}$ defined on a recombination search space or recombination P-structure $(X, \mathcal{R})$, where X is a finite set of all candidate solutions, and $\mathcal{R} : X \times X \to \mathcal{P}(X)$ is a certain finite interval operator mapping any pair of parents $(x, y) \in X \times X$ to their set of possible offspring $\mathcal{R}(x, y) \in \mathcal{P}(X)$ [139, 140, 150]. Having no equivalent definition via graphs or neighbourhood functions, recombination P-structures are thus inherently different from mutation search spaces (Chapter 4), and it explains why ELT

distinguishes recombination fitness landscapes from mutation fitness landscapes in line with the 'one operator one landscape' view [73]. Nevertheless, recombination fitness landscapes still admit a decomposition analogous to mutation fitness landscapes via a Fourier series expansion of their fitness function [139, 140, 150]. The next Section 10.3.1 explains why it is so.

10.3.1 Recombination P-structure Laplacian and the Mutation-recombination Homomorphism

Every fitness function $f \in \mathbb{R}^X$ can be linearly decomposed using Equation 10.2 if an orthonormal basis that spans $\mathbb{R}^X$ is known, regardless of whether f is defined on mutation or recombination fitness landscapes. The question is if such orthonormal basis is associated with the underlying structure of the fitness landscape at hand. Except in special cases explained later in this section, recombination fitness landscapes $(X, \mathcal{R}, f)$ are not decomposable with respect to the orthonormal basis of the graph Laplacian $\mathbf{L}$ because $\mathbf{L}$ itself is undefined for recombination P-structures $(X, \mathcal{R})$. Accordingly, investigations in ELT [139, 140, 150] began exploring a generalisation of the graph Laplacian for recombination P-structures. That is, a linear operator $\mathbf{L}_{\mathscr{R}} : \mathbb{R}^X \to \mathbb{R}^X$ called *recombination P-structure Laplacian matrix* with dimensions $|X| \times |X|$

$$\mathbf{L}_{\mathscr{R}} \stackrel{\text{def}}{=} 2\,|X|\,\mathbf{I} - \mathbf{S} \ , \tag{10.10}$$

for a recombination P-structure $\mathscr{R} = (X, \mathcal{R})$. The matrix $\mathbf{S}$ is the generalised adjacency matrix (see Equation 7.5 in Chapter 7) of the recombination P-structure hypergraph associated with $\mathscr{R}$, and $2\,|X|\,\mathbf{I}$ is a diagonal matrix with diagonal entries all equal to $2\,|X|$. Due to the spectral theorem for symmetric linear operators [58, 113], if $\mathbf{L}_{\mathscr{R}}$ is symmetric, then the eigenfunctions of $\mathbf{L}_{\mathscr{R}}$ form an orthonormal basis of $\mathbb{R}^X$ as desired. The problem is $\mathbf{S}$ may not be symmetric, and thus neither $\mathbf{L}_{\mathscr{R}}$, for general recombination P-structures; therefore, there are no general guarantees $\mathbf{L}_{\mathscr{R}}$ has such basis. However, the following Proposition 10.1 (adapted, for simplicity, from Stadler and Wagner's Proposition 3 [139]) states the basis does exist if recombination P-structures are restricted to *generously transitive recombination P-structures* (Definition 10.4), whose automorphisms (see Definition 10.3 and Figure 10.5) are inspired by those of generously transitive graphs [14].

Definition 10.3 (**Recombination P-structure automorphism** [139])**.** An *automorphism* of a recombination P-structure $(X, \mathcal{R})$ is any function $\phi : X \to X$

such that $\mathcal{R}\big(\phi(x), \phi(y)\big) = \{\phi(z) \mid z \in \mathcal{R}(x, y)\}$ for all $x, y \in X$. The set of all automorphisms on $(X, \mathcal{R})$ is denoted $\mathrm{Aut}(X, \mathcal{R})$.

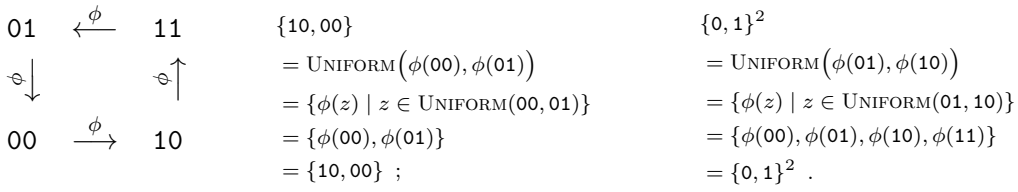


Figure 10.5. An automorphism ϕ for the uniform recombination P-structure $(\mathcal{H}_2^2, \textsc{Uniform})$ on two-dimensional binary Hamming sequences $\mathcal{H}_2^2$.

Definition 10.4 (**Generously transitive recomb. P-structure** [139])**.** A recombination P-structure $(X, \mathcal{R})$ is *generously transitive* if, for all $x, y \in X$, there exists an automorphism $\phi \in \mathrm{Aut}(X, \mathcal{R})$ such that $y = \phi(x)$.

Proposition 10.1 (**Existence of $\mathbf{L}_{\mathscr{R}}$ eigenbasis** [139])**.** For any generously transitive recombination P-structure $\mathscr{R} = (X, \mathcal{R})$, its generalised adjacency matrix $\mathbf{S}$ is symmetric, and thus the eigenfunctions of the recombination P-structure Laplacian $\mathbf{L}_{\mathscr{R}}$ constitute an orthonormal basis of $\mathbb{R}^X$.

Although string-based recombination P-structures (e.g. one-point and uniform) are the only known specific class whose generalised adjacency matrix $\mathbf{S}$ is symmetric [139], Proposition 10.1 is not necessarily limited to those in principle. Hence any recombination P-structure fitness landscape $(X, \mathcal{R}, f)$ has a Fourier series expansion with respect to the orthonormal basis eigenfunctions of $\mathbf{L}_{\mathscr{R}}$, as long as $\mathscr{R} = (X, \mathcal{R})$ is a generously transitive recombination P-structure. Then, an elementary landscape, akin to Definition 10.2 for mutation fitness landscapes, would be any such $(X, \mathcal{R}, f)$ whose Fourier series expansion consists of exactly one eigenfunction of $\mathbf{L}_{\mathscr{R}}$.

Definition 10.5 (**Recomb. P-structure elementary landscape** [139])**.** Let a fitness landscape $\mathscr{F}$ be defined by a fitness function $f : X \to \mathbb{R}$ on a generously transitive recombination P-structure $\mathscr{R} = (X, \mathcal{R})$. $\mathscr{F}$ is *elementary*[5] if and only if f is a recombination P-structure Laplacian eigenfunction φ_p with eigenvalue $\lambda_p \geq 0$ of order $p \geq 0$, so that: $(\mathbf{L}_{\mathscr{R}} f)(x) = \lambda_p f(x)$, $\forall x \in X$.

[5]To avoid awkward wordiness, the term 'recombination P-structure EL' is sometimes referred as 'EL'. The context will make clear whether EL refers to recombination P-structures or graphs.

Orthonormal Eigenbasis Equivalence between $\mathbf{L}$ and $\mathbf{L}_{\mathscr{R}}$

For generously transitive recombination P-structures $\mathscr{R}$, Proposition 10.1 guarantees the existence of an orthonormal basis spanning $\mathbb{R}^X$ that consists of $\mathbf{L}_{\mathscr{R}}$ eigenfunctions, but it does not indicate how to find them. Finding eigenfunctions analytically is already difficult for graph Laplacians $\mathbf{L}$ in general, despite there exist methodologies for that purpose [20], even more so for $\mathbf{L}_{\mathscr{R}}$. Certainly, the Walsh basis for pseudo-Boolean fitness functions in $\mathbb{R}^{\{0,1\}^n}$ is associated with $\mathbf{L}$ under the single bit-flip neighbourhood (Section 10.2.1); however, not all neighbourhoods or search spaces are associated with a Walsh basis. For example, there is no Walsh basis for permutations in the TSP [125]. Moreover, the bases of $\mathbf{L}$ and $\mathbf{L}_{\mathscr{R}}$ need not be the same, and investigating the algebraic properties of $\mathbf{L}_{\mathscr{R}}$, besides symmetry and having non-negative real eigenvalues, poses additional complications [139]. Effectively, for generously transitive recombination P-structures there is little guidance on how to find a basis when it is unknown.

To mitigate those difficulties, Proposition 10.2 below (see Corollary 1 in [139]) goes a step further than Proposition 10.1 and provides a sufficient condition where the eigenfunctions of $\mathbf{L}$ and those of $\mathbf{L}_{\mathscr{R}}$ are identical; that is, where both share exactly the same orthonormal eigenbasis spanning $\mathbb{R}^X$ [139]. This condition is *distance-transitivity* on the backbone graphs associated with recombination P-structures (Definition 10.6) analogous to the standard notion of distance-transitivity in graph theory [14], and, as explained next, it relates directly to the mutation-recombination homomorphism seen in Chapter 4.

Definition 10.6 (**Backbone distance-transitive recomb. P-structure** [139])**.** Let $\mathscr{R} = (X, \mathcal{R})$ be a recombination P-structure with a connected backbone graph $\mathrm{bbg}_{\mathscr{R}} = (V, E)$ and whose natural shortest-path metric is d, and $\mathrm{Aut}(X, \mathcal{R})$ the set of automorphisms of $\mathscr{R}$. Then, $\mathscr{R}$ is *distance-transitive* with respect to $\mathrm{bbg}_{\mathscr{R}}$, if for every two vertex pairs $(x, y), (u, v) \in V \times V$, with $d(x, y) = d(u, v)$, there is an automorphism $\phi \in \mathrm{Aut}(X, \mathcal{R})$ such that $x = \phi(u)$ and $y = \phi(v)$.

Proposition 10.2 (**Eigenfunctions of $\mathbf{L}$ and $\mathbf{L}_{\mathscr{R}}$ are identical** [139])**.** If a recombination P-structure $\mathscr{R} = (X, \mathcal{R})$ is both backbone distance-transitive and generously transitive, then the eigenfunctions of the recombination P-structure Laplacian $\mathbf{L}_{\mathscr{R}}$ are identical to the eigenfunctions of the graph Laplacian $\mathbf{L}$ for the backbone graph associated with $\mathscr{R}$.

Proposition 10.2 Implications: Mutation-recombination Homomorphism

Proposition 10.2 implies that if a fitness function f is elementary for a certain graph G as in Definition 10.2, then f is elementary as in Definition 10.5 for any recombination P-structure $(X, \mathcal{R})$ conforming to Proposition 10.2 that has G as its backbone graph. Furthermore, recall from Section 10.1.1 that, once a basis is fixed, to every $f \in \mathbb{R}^X$ there corresponds a unique real vector of coefficients in $\mathbb{R}^{|X|}$ that linearly combined with the basis functions generate f. So the Fourier series expansion of f is the same with respect to $(X, \mathcal{R})$ and G above for sharing the same basis, assuming that G has vertex set $V = X$. None of this, however, necessarily means that the mutation (G, f) and recombination $(X, \mathcal{R}, f)$ landscapes themselves are identical in every aspect. Stadler and Wagner [139] show that (G, f) and $(X, \mathcal{R}, f)$ possibly have, for example, different correlation or ruggedness because their associated eigenvalues may be different even if Proposition 10.2 holds.

What makes Proposition 10.2 significant is allowing us to compare (possibly different) ELs and recombination P-structure ELs with respect to a common eigenbasis as long as Proposition 10.2 is satisfied. This comparability is rooted in the underlying mutation-recombination homomorphisms [26, 51] known in ELT [139], whereby the graph induced by neighbourhoods of typical mutation operators is (up to isomorphism) the backbone graph embedded in the hypergraph of recombination P-structures, and this embedding is a homomorphism from the backbone graph to the hypergraph.

A trivial example where Proposition 10.2 holds is the n-bit flip neighbourhood $\textsc{Bitflip}(n)$ and the identity recombination P-structure $(\{0,1\}^n, \textsc{Id})$ seen in Chapter 4. The reason being that the hypergraph and backbone graph of $(\{0,1\}^n, \textsc{Id})$ are both isomorphic to the complete graph K_{2^n} induced by $\textsc{Bitflip}(n)$.

A non-trivial example is the binary Hamming graph induced by the single bit-flip neighbourhood, which is the backbone graph embedded in the hypergraph of the uniform recombination P-structure associated with uniform crossover (Chapter 4). The uniform recombination P-structure[6] fulfils Proposition 10.2, and therefore its recombination P-structure Laplacian shares the same eigenfunctions as the Hamming graph Laplacian according to Corollaries 2–3 in [139]: these are exactly Walsh functions (Section 10.2.1), extended to non-binary Hamming graphs as well in [140, 150].

To summarise, Proposition 10.2, alongside mutation-recombination homomorphisms, is useful to analyse or find ELs for existing or new recombination P-structures because:

[6]Its backbone distance-transitivity is basically due to Hamming graphs' distance-transitivity [14].

- The graph Laplacian eigenfunctions of many NP-complete problems are known already for common neighbourhood functions [5, 20, 56, 85, 136, 137].
- The problem of finding an orthonormal basis of recombination P-structure Laplacians $\mathbf{L}_{\mathscr{R}}$ (Equation 10.10) reduces to the problem of finding an orthonormal basis of graph Laplacians $\mathbf{L}$ (Equation 10.4) for the backbone graph associated with $\mathscr{R}$, where the neighbourhood and the basis are possibly known or can be less difficult to find.
- In essence, recombination P-structures fulfilling Proposition 10.2 do not have to be directly dealt with to determine what their ELs are: the graph Laplacian eigenfunctions of the backbone graph embedded in their hypergraphs suffice.

Elementary Landscapes via Headless-chicken Crossover Random Walks

Recombination P-structure ELs (Definition 10.5) may be defined in terms of transition probability matrices $\mathbf{P}$ for headless-chicken (HC) crossover random walks introduced in Chapter 7. Particularly, for HC crossover random walks that assume a uniform probability distribution over the finite set X of all states (i.e. candidate solutions), Equation 7.4 and Equation 10.10 relate $\mathbf{P}$ and recombination P-structure Laplacians $\mathbf{L}_{\mathscr{R}}$ through the identity

$$\mathbf{P} = \frac{1}{2\,|X|}\,\mathbf{S} = \mathbf{I} - \frac{1}{2\,|X|}\,\mathbf{L}_{\mathscr{R}} \quad . \tag{10.11}$$

As pointed out in [139, 150], $\mathbf{P}$ and $\mathbf{L}_{\mathscr{R}}$ have the same eigenfunctions provided that $\mathbf{S}$ is symmetric. Therefore, a recombination fitness landscape is *elementary* also when its fitness function is an eigenfunction of the transition probability matrix of a HC crossover random walk on that landscape.

When HC crossover random walks are based on other (non-uniform) probability distributions, such as linkage equilibrium [50] (see Section 7.3.3 in Chapter 7), it is possible to define 'weighted' versions of $\mathbf{P}$ and $\mathbf{L}_{\mathscr{R}}$ accounting for frequencies of individuals in a population modelled by the distribution [140]. However, doing so implies that a recombination fitness landscape may or may not be elementary depending on which distribution is used.

10.4 Structure of Elementary Fitness Landscapes

Sections 10.2–10.3 presented a general approach to decompose mutation fitness landscapes into a sum of ELs and recombination fitness landscapes into a sum of re-

combination P-structure ELs. However, due to the distinct nature of graphs and recombination P-structures, the analysis of ELs and recombination P-structure ELs requires separate approaches in general. For the former, spectral graph theory is readily available; for the latter, ELT envisions a generalised spectral graph theory via recombination P-structure Laplacians to be developed. Fortunately, the existence of mutation-recombination homomorphisms and Proposition 10.2 provide conditions where recombination P-structure ELs and ELs are comparable in a fundamental sense (Section 10.3.1). The remainder of this chapter introduces a general approach to analyse the structure of ELs and, as long as the aforesaid conditions hold, may be extensible to recombination P-structure ELs through their backbone graphs.

10.4.1 Discrete Nodal Domains and Local Optima

The graph Laplacian spectrum (Equation 10.5) of an EL bears intimate and hidden relationships with the structure of its local optima. To reveal and understand these relationships, ELT [85, 136, 137, 139] suggests *discrete nodal domains* DNDs, also called *sign graphs* for they represent connected subsets of vertices where image values of eigenfunctions do not change sign, according to Definition 10.7 next. Figure 10.6 exemplifies the four possible types of DNDs.

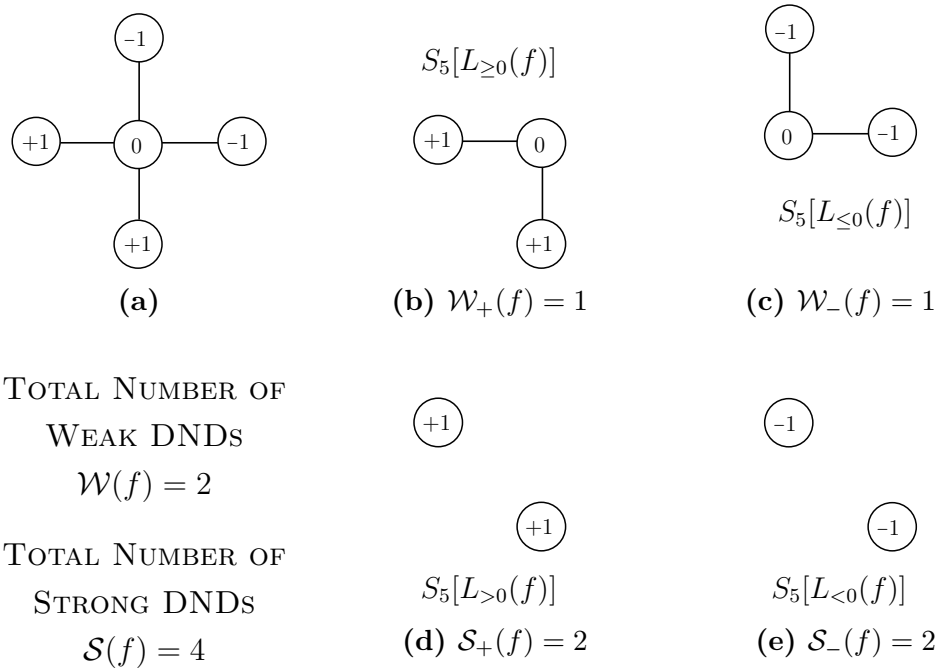


Figure 10.6. A function $f : V \to \{-1, 0, +1\}$ defined on the vertex set V of a star graph S_5 (Figure 10.6a) and its associated discrete nodal domains: the positive weak and negative weak DNDs in Figures 10.6b–10.6c, and, the positive strong and negative strong DNDs in Figures 10.6d–10.6e.

Definition 10.7 (**Discrete nodal domains** [**DNDs**] [8, 30])**.** Let $f : V \to \mathbb{R}$ be a graph Laplacian eigenfunction defined on the vertex set V of a finite connected graph G. A *positive weak* DND is a maximal connected subgraph of G, denoted $G[L_{\geq 0}(f)]$, induced by the super-level set $L_{\geq 0}(f) = \{v \in V \mid f(v) \geq 0\}$ with one non-zero vertex at least. *Negative weak*, *positive strong* and *negative strong* DNDs are defined analogously by replacing the previous inequalities $\geq$ with either $\leq$, $>$ or $<$ respectively. The *number* of these DNDs is denoted $\mathcal{W}_+(f)$, $\mathcal{W}_-(f)$, $\mathcal{S}_+(f)$ and $\mathcal{S}_-(f)$ respectively. The *total number* of weak and strong DNDs of f are $\mathcal{W}(f) = \mathcal{W}_+(f) + \mathcal{W}_-(f)$ and $\mathcal{S}(f) = \mathcal{S}_+(f) + \mathcal{S}_-(f)$.

Another way to refer to DNDs are the connected components given by pre-images of the fitness function defined on a graph. For example, on the star graph S_5 in Figure 10.6a, the connected component of $f^{-1}\Big((-\infty, 0]\Big)$ is exactly the subgraph $S_5[L_{\leq 0}(f)]$ shown in Figure 10.6c, which is a negative weak DND according to Definition 10.7. Likewise, $f^{-1}\Big((0, +\infty)\Big)$ has two connected components corresponding to the subgraph $S_5[L_{>0}(f)]$, namely two positive strong DNDs that happen to be two isolated vertices as shown in Figure 10.6d.

Besides their presence in important combinatorial optimisation problems, ELs are an interesting class of fitness landscapes because their local optima exhibit a distinctive structure that manifests in their DNDs and justifies why ELT regards ELs as 'well behaved' compared with non-elementary fitness landscapes [85]. That is, the *maximum principle* of non-flat ELs in Proposition 10.3 consisting of various inequalities (see Proposition 3 and Corollary 4 in [35], and Theorem 5 in [5]).

Proposition 10.3 (**Maximum principle of non-flat ELs** [5, 35, 85])**.** Let $c \in \mathbb{R}$ and a fitness function $f : V \to \mathbb{R}$ defined on a finite connected graph $G = (V, E)$. If $f' = f - c\mathbf{1}$ is an eigenfunction of the graph Laplacian $\mathbf{L}$ on G with eigenvalue $\lambda > 0$, then for all local minima $v_{\min} \in V$ and all local maxima $v_{\max} \in V$:

$$f'(v_{\min}) \leq c \leq f'(v_{\max}) \ . \tag{10.12}$$

Specifically, if f' is an eigenfunction for the offset $c = \bar{f} = \frac{1}{|V|}\sum_{v \in V} f(v)$, then:

$$\bar{f} - \frac{\varepsilon^*}{\lambda} \leq f'(v_{\min}) \leq \bar{f} \leq f'(v_{\max}) \leq \bar{f} + \frac{\varepsilon^*}{\lambda} \ , \tag{10.13}$$

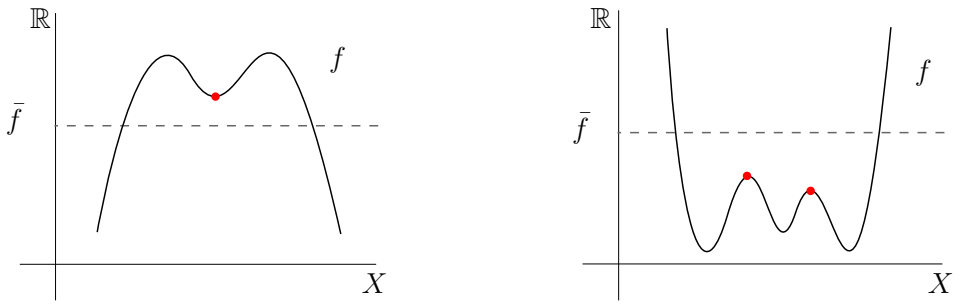
where $\varepsilon^* = \max_{\{u,w\} \in E} |f'(u) - f'(w)|$ is the maximum local variation in fitness.

Remark 10.3 (Proposition 10.3)**.** The original maximum principle considered by Grover [56] and Stadler [136, 137] is stated specifically for zero-averaged functions

$f - \bar{f}\mathbf{1}$ with offset $c = \bar{f} = \frac{1}{|V|}\sum_{v \in V} f(v)$. Later, Barnes and others [5] extended it to fitness functions with any offset as in Equation 10.12 (see Remark 10.1).

Example 10.2 (**Maximum principle on star graph** S_5)**.** Consider the function f defined on the vertex set of the star graph S_5 in Figure 10.6a. Let the offset $c = \bar{f}$, so $f' = f - \bar{f}\mathbf{1} = f$ since the global average of f is $\bar{f} = 0$. The maximum local variation is $\varepsilon^* = 1$. From the example in Figure 10.2 earlier in Section 10.2.1: $\mathbf{L}f = \lambda f$ with $\lambda = 1$ of order 1 since $Spectrum(S_5) = \{0_{[1]}, 1_{[3]}, 5_{[1]}\}$. Substituting in Equation 10.13: $-1 \leq f(v_{\min}) \leq 0 \leq f(v_{\max}) \leq 1$. So it is clear from Figure 10.6a that the maximum principle holds for all local minima $v_{\min}$ and local maxima $v_{\max}$.

For the maximum principle, therefore, ELs have all local maxima in positive weak (or strong) DNDs and all local minima in negative weak (or strong) DNDs; local optima include global optima as well. Particularly, choosing the offset $c = \bar{f}$ makes the zero fitness level of $f' = f - \bar{f}\mathbf{1}$ coincide with the global average $\bar{f}$ fitness level, and thus positive DNDs would contain candidate solutions of above-average fitness, whereas negative DNDs would contain those of below-average fitness. Furthermore, the fitness of any locally optimal solution will lie in the interval $[\bar{f} - \frac{\varepsilon^*}{\lambda}, \bar{f} + \frac{\varepsilon^*}{\lambda}]$ due to the lower and upper bounds in Equation 10.13 of the maximum principle.



(a) Local minimum above average $\bar{f}$ **(b)** Local maxima below average $\bar{f}$

Figure 10.7. Informal sketch of fitness landscapes that cannot be elementary landscapes due to the maximum principle (Proposition 10.3).

The number of DNDs (regardless of which type) in ELs depends on the graph and function at hand and, unlike the simple example in Figure 10.6, counting them is intractable in general [8]. Nevertheless, the *discrete nodal domain theorem* [8, 30] in Proposition 10.4, which is a discrete version of Courant's nodal domain theorem for eigenfunctions on continuous spaces [25], provides general upper-bounds on the total number of weak and strong DNDs of an EL. This section focuses on graph Laplacians, so Proposition 10.4 and the improved upper-bounds in Proposition 10.5

are specific instances of the original statements in [8, 30] expressed more generally with *generalised Laplacians* or *essentially non-positive matrices* of graphs.

Proposition 10.4 (**General upper-bounds DNDs** [8, 30])**.** Given a connected graph with n vertices, any graph Laplacian eigenfunction f_k corresponding to the k-th eigenvalue λ_k, with multiplicity r, has $\mathcal{W}(f_k) \leq k$ and $\mathcal{S}(f_k) \leq k + r - 1$.

As soon as a fitness function is known to be elementary, the position and multiplicity of the corresponding eigenvalue in the spectrum are known. Indeed, closed-form analytical expressions of graph Laplacian eigenfunctions and eigenvalues can be derived for many important combinatorial optimisation problems [20, 56, 85, 136, 137]. Thus, if we think of DNDs as 'clusters' of local optima in ELs, Proposition 10.4 provides a direct analytical way to upper-bound the total number of those clusters solely from the graph Laplacian spectrum (Equation 10.5); no further assumptions nor experimental data on ELs are required. This could be useful to design EAs or guide their search. The actual number of local optima in an EL may exceed the number of DNDs though: DNDs can contain multiple local optima (Figures 10.6b–10.6c). The upper-bounds can be improved for certain classes of graphs such as binary Hamming graphs as in Proposition 10.5, which are sharp for eigenvalue orders $p \in \{0, 1, 2\}$ regardless of the number of vertices (see Table 4.1 in [8]).

Proposition 10.5 (**Upper-bounds on hypercubes: weak DNDs** [8])**.** Let f_p be a graph Laplacian eigenfunction, with eigenvalue $2p$, for a n-dimensional binary Hamming graph $H(n, 2)$, where p is the eigenvalue order (ignoring multiplicities). Let also N be the number of vertices of $H(n, 2)$. Then, the total number $\mathcal{W}(f_p)$ of weak DNDs is upper-bounded as:

$$\mathcal{W}(f_p) \leq w_{N,p} = \begin{cases} p + 1 & \text{if } p = 0 \text{ or } p = 1, \\ 2\left(1 + \sum_{k=0}^{p/2-1} \binom{N}{2k}\right) & \text{if } p \text{ is even}, \\ 2\left(1 + \sum_{k=0}^{\lfloor p/2 \rfloor - 1} \binom{N}{2k+1}\right) & \text{if } p \text{ is odd}. \end{cases} \tag{10.14}$$

ELs corresponding to the smallest non-zero eigenvalue (of order one) are a special class because the total number of weak DNDs is exactly two; see Proposition 10.6 below, or Corollary 3.2 in [8] and Theorem 3.3 in [44]. Motivated by Kauffman [77], Stadler calls the order-one ELs *Fujiyama*[7] for they happen to have a unique local

[7] Fujiyama ELs are called also *Fiedler vectors* or *characteristic valuations* [8, 44].

maximum and minimum on binary Hamming graphs (not so for other graphs necessarily) [136, 137, 139]. By contrast, *non-Fujiyama* ELs possibly have more than two weak DNDs.

Proposition 10.6 (**Fujiyama ELs: 2 weak DNDs in total** [8, 44])**.** A graph Laplacian eigenfunction f corresponding to the smallest non-zero eigenvalue of any connected graph has $\mathcal{W}(f) = 2$ weak DNDs.

A function defined in Euclidean space where all sub-level sets' pre-images have a single connected component is *unimodal* according to Kanemitsu and others [75]. Proposition 10.7 (see Corollary 2.6 in [145]) states that, for certain graphs, Fujiyama ELs satisfy a notion of unimodality, or topological monotonicity [43], given in Definition 10.8 generalising what Tlusty calls *tightness* [145].

Definition 10.8 (**Weakly unimodal / tight function**)**.** Let an arbitrary real-valued function f defined on the vertex set of a finite connected graph G. Then, f is called *weakly unimodal* or *weakly tight* on G if the subgraphs induced by all sub-level *or* super-level sets on G are connected, namely $G[L_{\leq \ell}(f)]$ *or* $G[L_{\geq \ell}(f)]$ or both are connected $\forall \ell \in \mathbb{R}$. If $G[L_{\leq \ell}(f)]$ *and* $G[L_{\geq \ell}(f)]$ are connected $\forall \ell \in \mathbb{R}$, then f is called *tight*.

Proposition 10.7 (**Fujiyama ELs: unimodality/tightness** [145])**.** Let f be a real-valued graph Laplacian eigenfunction corresponding to the smallest non-zero eigenvalue for any connected graph G resulting from a Cartesian product of paths, cycles, or complete graphs. Then, the subgraphs $G[L_{\leq \ell}(f)]$ *and* $G[L_{\geq \ell}(f)]$ induced, respectively, by all sub-level *and* super-level sets of f on G are connected $\forall \ell \in \mathbb{R}$.

Remark 10.4 (To Proposition 10.7)**.** In general, neither $G[L_{\leq \ell}(f)]$ nor $G[L_{\geq \ell}(f)]$ need to be connected for non-constant graph Laplacian eigenfunctions on arbitrary connected graphs [145].

Example 10.3 (**Weakly unimodal / tight Fujiyama EL**)**.** The function f defined on the star graph S_4 shown in Figure 10.6 is a weakly unimodal Fujiyama EL but not tight: notice $S_4[L_{\geq -1}(f)]$ is a connected subgraph but $S_4[L_{\leq -1}(f)]$ is disconnected, also $S_4[L_{\leq +1}(f)]$ is connected but $S_4[L_{\geq +1}(f)]$ is disconnected. Any of the order-one Walsh functions on the binary Hamming graph shown in Figure 10.3 are tight Fujiyama ELs (thus also weakly tight or weakly unimodal). Constant functions defined on connected graphs are obviously Fujiyama ELs and also (weakly) tight since the use of inequalities in Definition 10.8 allows subsets of connected vertices with equal fitness.

Compared with general Fujiyama ELs, weakly unimodal Fujiyama ELs (and also tight ones) are a more restricted class. Both have two weak DNDs in total, where all local minima belong to a negative weak DND and all local maxima belong to a positive weak DND due to the maximum principle (Proposition 10.3). However, in a weakly unimodal Fujiyama EL all subgraphs $G[L_{\leq \ell}(f)]$ or $G[L_{\geq \ell}(f)]$ are connected for all fitness co-domain values $\ell \in \mathbb{R}$, whereas in a general Fujiyama EL that is not necessarily true (except for $\ell = 0$ due to Proposition 10.6). Put differently, each level ℓ represents a hyperplane that always 'cuts' a weakly unimodal Fujiyama EL into no more than two connected components, not so for other ELs in general.

Fitness landscapes in ELT, therefore, belong to one of the following mutually-exclusive classes: not elementary, elementary and flat, elementary and not flat. The latter can be further subdivided into: tight Fujiyama, weakly unimodal (or weakly tight) Fujiyama, Fujiyama, and non-Fujiyama (which are generally multi-modal).

Chapter 11

Abstract Convex Elementary Landscapes

This chapter is an original major contribution of this thesis, extending a paper that I co-authored with Moraglio [48]. This chapter presents theoretical foundations to systematically classify combinatorial fitness landscapes, across problems and representations, with respect to the classes of abstract convex landscapes and elementary landscapes reviewed in Chapters 9 and 10 respectively.

11.1 Introduction

The crossover classification presented in Chapter 5 laid the foundations to develop in Chapter 8 a qualitative framework for analysing the abstract behaviour of EAs, based on geometric crossovers and recombination P-structures, across problems and representations. What about their performance? To explain why some EAs perform better on some problems and worse on others, it is necessary to understand how search operators driving EAs' behaviour relate to the problems' fitness functions. Fitness landscapes formalise that relationship and their classification [69], according to key landscape features affecting the performance of a class of EAs, is a means to separate which classes of problems may be 'easier' or 'harder' for those EAs.

On certain classes of abstract convex landscapes with a globally convex or big valley structure [101], formal evidence in the GF [100, 101, 104] supports that geometric-crossover EAs could perform exponentially better than pure random search at finding optimal or 'good' suboptimal solutions. However, it is unclear what specific landscapes those classes contain: is it real-world problems, classical combinatorial optimisation problems or simple benchmark problems like leading-ones?

Could we show a priori and systematically when a landscape belongs to any of those classes? Answering these questions would be a major leap in the GF to formally explain which geometric-crossover EA performs best on which problem.

To invoke ELT for an answer may seem contrived since ELT does not focus on the analysis of globally convex functions. Nevertheless, elementary landscapes [136] represent relevant fitness landscapes of real phenomena like spin-glass models of magnetic materials, of software engineering like the next release problem [91], and of various combinatorial NP-complete problems. Some of these elementary landscapes [138], for instance those related with TSP or graph bipartition problems, also exhibit a big valley or globally convex structure apparently due to an underlying ultrametric distance [10, 46, 83]. Furthermore, because the fitness function of elementary landscapes is an eigenfunction of a graph Laplacian or recombination P-structure Laplacian operator, valuable information about their local optima and features like smoothness can be analytically derived from the Laplacian spectrum [136, 139]. What still remains obscure about elementary landscapes is the relation between their geometrical structure, spectral properties and population-based EAs [85] as opposed to local search [56] and recombination P-structure random walks [140].

Pursuing a classification of fitness landscapes that are both abstractly convex and elementary would help us clarify if: (a) the abstract convex landscape classes contain instances of interesting elementary landscapes; (b) there exist elementary landscapes where geometric-crossover EAs potentially outperform pure random search; and, (c) the abstract convexity of elementary landscapes can be inferred analytically from the corresponding Laplacian spectrum. But it is far from clear if such classification is possible at all; abstract convex landscapes and elementary landscapes, as introduced in Chapters 9 and 10, are formally different classes of fitness landscapes for which no unified theory has been developed yet.

The aim of this chapter is to explore, based on prior research I co-authored [48], a mathematical characterisation of landscapes that are both abstractly convex and elementary. That is, to develop theoretical foundations of abstract convex elementary landscapes by focusing on the following two fundamental questions framed in Figure 11.1 below.

Question 11.1. *Are the classes of elementary landscapes in ELT and abstract convex fitness landscapes in the GF disjoint? If not, which specific classes do overlap?*

Question 11.2. *How does the abstract convexity of an elementary landscape relate to its eigenfunction order or position in the Laplacian spectrum?*

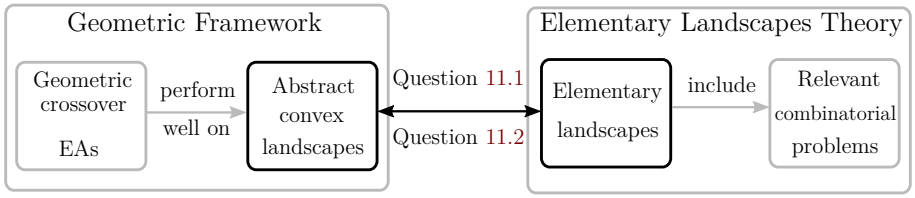


Figure 11.1. General view of central questions and topics (highlighted in 'black') in Chapter 11 with respect to the geometric framework and elementary landscapes theory.

11.1.1 Overview

This chapter proves abstract convex landscapes and elementary landscapes are not disjoint classes, though sometimes become degenerate and include all landscapes or flat landscapes only. Also, it proves the abstract convexity present in certain abstract convex elementary landscapes can be directly identified by looking at the position in the graph Laplacian spectrum. To do so, this chapter begins addressing Question 11.1 with the definition of various pseudo-Boolean fitness landscapes (Section 11.2) and their classification (Section 11.3) to find which ones are abstract convex elementary landscapes. Later, Section 11.4.1 focuses on Question 11.2 and proves that (quasi-)convex or (quasi-)concave elementary landscapes generally have eigenvalues of order less than two. Finally, Section 11.4.2 shows there exist approximately (average or quasi) convex elementary landscapes and explains how they relate to the Laplacian spectrum. All together will reveal important implications, discussed in Section 11.5, concerning the viability of classifying abstract convex elementary landscapes to identify the kind of combinatorial optimisation problems where certain geometric-crossover or recombination P-structure EAs perform best.

11.1.2 Contributions

The main contributions of this chapter are:

1. A classification of pseudo-Boolean fitness landscapes, defined on finite metric spaces with the discrete metric or Hamming metric, associated with the following problems: one-max, needle-in-a-haystack, leading-ones, not-all-equal 3-satisfiability, and weight partitioning. (Section 11.3.)

2. A proof that every non-constant fitness function on complete graphs belongs to the classes of: (approximately) convex, (approximately) quasi-convex, (approximately) average-convex, and tight Fujiyama elementary landscapes, as well as their concave versions. (Section 11.3.1 and Section 11.4.2.)

3. A proof that, on binary Hamming graphs, tight Fujiyama elementary landscapes overlap with (approximately) average-convex and (approximately) average-concave landscapes using one-max and certain perturbed one-max fitness landscapes as examples. (Section 11.3.2 and Section 11.4.2.)

4. A proof that for any finite connected graph, if an elementary landscape is (quasi-)convex or (quasi-)concave, then it corresponds to graph Laplacian eigenfunction with an eigenvalue of order less than two. (Section 11.4.1.)

5. A proof that for the class of triangle-free graphs of vertex degree greater or equal than two, which includes all non-trivial binary Hamming graphs, there is no convex nor concave elementary landscape except the constant order-zero graph Laplacian eigenfunction for eigenvalue zero. (Section 11.4.1.)

6. Characterising perturbed fitness landscapes that are elementary as well as approximately (average or quasi) convex. (Section 11.4.2.)

Section 11.5 discusses the contributions, and Section 11.6 concludes this chapter.

11.2 Preliminaries

Pseudo-Boolean fitness landscapes must be defined before classifying them. The next Section 11.2.1 defines pseudo-Boolean optimisation problems, and Section 11.2.2 defines pseudo-Boolean fitness landscapes associated with those problems.

11.2.1 Pseudo-Boolean Optimisation Problems

This chapter considers five pseudo-Boolean optimisation problems with distinct computational complexities and characteristics, which previous studies [87, 104, 136, 137] and one I co-authored [48] have analysed with regards to abstract convex landscapes and elementary landscapes: one-max, needle-in-a-haystack, leading-ones, not-all-equal 3-satisfiability problem (NAE3SATP), and weight partitioning problem (WPP). The first three (Definitions 11.1–11.3) are benchmark problems whose analysis can be insightful despite their simplicity and which, except for needle-in-a-haystack, are solvable in polynomial time [3, ch. 2]. By contrast, NAE3SATP and WPP are two of Karp's classical 21 NP-complete problems [56, 76]: the former is a version of Boolean satisfiability restricted to three-literal clauses (Definition 11.4), and the latter is a version of two-way number partitioning (Definition 11.5).

Definition 11.1 (**One-max problem** [3])**.** The *one-max problem* is to find the n-dimensional binary Hamming sequence, where $n \in \mathbb{N}$, that maximises the *one-max fitness function*

$$\begin{aligned} f_{\textsc{OneMax}} : \{0,1\}^n &\to \{0,\dots,n\} \\ (x_1,\dots,x_n) &\mapsto \sum_{i=1}^{n} x_i \ . \end{aligned}$$

Definition 11.2 (**Needle-in-a-haystack problem** [3])**.** The *needle-in-a-haystack problem* is to find the n-dimensional binary Hamming sequence, where $n \in \mathbb{N}$, that maximises the *needle fitness function*

$$\begin{aligned} f_{\textsc{Needle}} : \{0,1\}^n &\to \{0,1\} \\ (x_1,\dots,x_n) &\mapsto \prod_{i=1}^{n} x_i \ . \end{aligned}$$

Definition 11.3 (**Leading-ones problem** [3])**.** The *leading-ones problem* is to find the n-dimensional binary Hamming sequence, where $n \in \mathbb{N}$, that maximises the *leading-ones fitness function*

$$\begin{aligned} f_{\textsc{LeadingOnes}} : \{0,1\}^n &\to \{0,\dots,n\} \\ (x_1,\dots,x_n) &\mapsto \sum_{i=1}^{n} \prod_{j=1}^{i} x_j \ . \end{aligned}$$

Definition 11.4 (**Not-all-equal 3-satisfiability problem** [**NAE3SATP**] [56])**.** Let an n-dimensional sequence of binary variables $x = (x_1,\dots,x_n)$ and their complements $\bar{x}_i = 1 - x_i$ define a set of literals $L = \{x_1, \bar{x}_1, x_2, \bar{x}_2, \dots, x_n, \bar{x}_n\}$ for $n \in \mathbb{N}$. An *instance* consists of a finite family $\mathcal{C}_x$ of subsets $C \subseteq L$ called clauses such that: each clause C has three literals, and either $x_i \in C$ or $\bar{x}_i \in C$ not both. A clause is satisfied if and only if at least one of its literals evaluates to 0 and another to 1. The *not-all-equal 3-satisfiability problem* (NAE3SATP) is to find a $(x_1,\dots,x_n) \in \{0,1\}^n$ maximising the number of satisfied clauses, that is the NAE3SATP *fitness function*

$$\begin{aligned} f_{\textsc{NAE3SATP}} : \{0,1\}^n &\to \{0,\dots,|\mathcal{C}_x|\} \\ x = (x_1,\dots,x_n) &\mapsto \sum_{C \in \mathcal{C}_x} \underbrace{\left(1 - \prod_{c \in C} c\right)\left(1 - \prod_{c \in C} \bar{c}\right)}_{\text{is 1 iff } x \text{ satisfies } C\text{, otherwise 0}} \ . \end{aligned}$$

Definition 11.5 (**Weight partitioning problem** [**WPP**] [56])**.** An *instance* con-

sists of an n-dimensional sequence $(w_1, \ldots, w_n)$ of real numbers representing 'weights' corresponding to 'objects' $(o_1, \ldots, o_n)$ for $n \in \mathbb{N}$. The *weight partitioning problem* (WPP) is to find a $(x_1, \ldots, x_n) \in \{0,1\}^n$ that minimises the WPP *fitness function*

$$f_{\text{WPP}} : \{0,1\}^n \to \mathbb{R}$$
$$f_{\text{WPP}}(x_1, \ldots, x_n) \mapsto \left(\sum_{i=1}^{n} w_i \cdot s(x_i) \right)^2 ,$$

where $s(x_i) = -1$, if $x_i = 0$, otherwise $s(x_i) = 1$.

11.2.2 Pseudo-Boolean Fitness Landscapes

To achieve a classification of pseudo-Boolean fitness landscapes that helps us address Questions 11.1 and 11.2 we seek pseudo-Boolean fitness landscapes that conceivably lie in the intersection, visualised in Figure 11.2, between the abstract convex landscapes, elementary landscapes and recombination P-structure elementary landscapes classes seen in Chapters 9 and 10.

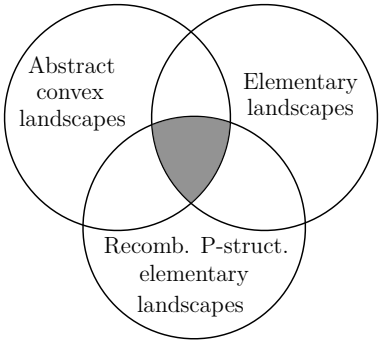


Figure 11.2. Venn diagram highlighting the subclass that intersects abstract convex and (recombination P-structure) elementary landscapes.

Because (pseudo-Boolean) fitness landscapes generally may not be in that intersection, we impose three minimum requisites to exclude landscapes that surely are not in the intersection:

- To be finite. Abstract convex landscapes may be continuous or finite, but (recombination P-structure) elementary landscapes must be finite always.
- To be associated with a metric. Abstract convex landscapes are always associated with a metric and elementary landscapes based on neighbourhoods of graphs too (e.g. shortest-path metric); recombination P-structure elementary landscapes can but generally are not associated with metrics.

- To be elementary with respect to a neighbourhood and a recombination P-structure because elementary landscapes need not be elementary in both senses. (The GF does not define mutation and recombination landscapes separately; here we enforce ELT to make no formal distinction between them neither.)

The above criteria is met by restricting to fitness landscapes defined on finite and connected backbone graphs that are homomorphically equivalent to hypergraphs of recombination P-structures satisfying generous transitivity and backbone distance-transitivity (see Section 10.3.1 in Chapter 10). Here Proposition 10.2 (see Corollary 1 by Stadler and Wagner [139]) guarantees that any fitness function elementary for the neighbourhoods of such graphs is also elementary for the corresponding generous and backbone distance-transitive recombination P-structures and vice versa.

Out of all crossovers classified in Chapters 5 and 8, the only provably known recombination P-structures with such properties and backbone graphs are the trivial case of complete graphs for identity crossover and the non-trivial case of binary Hamming graphs for unbiased uniform crossover (see Section 10.3.1 and [139, 150]). Other crossovers may well be generous and backbone distance-transitive recombination P-structures, but proving that involves research out of this chapter's scope.

Complete graphs and binary Hamming graphs are associated with the metric spaces, neighbourhoods and geometric recombination P-structures specified in Table 11.1 below.

Metric space		Graph	Neighbourhood function	Geometric recombination P-structure
Set	Metric			
$\{0,1\}^n$	discrete metric d_0	complete graph K_{2^n}	n-bit flip	$(\{0,1\}^n, \textsc{Id})$
$\{0,1\}^n$	Hamming metric d_{H}	binary Hamming graph $H(n,2)$	single-bit flip	$(\{0,1\}^n, \textsc{Uniform})$

Table 11.1. Two kinds of graphic metric spaces alongside associated neighbourhoods and geometric-recombination P-structures satisfying Proposition 10.2.

Definition 11.6 (**Discrete-distance metric space** [34])**.** A *discrete-distance metric space*, for short *d_0-metric space*, (X, d_0) consists of an arbitrary set X and the *discrete metric* d_0 defined $\forall x, y \in X$ as: $d_0(x,y) = 0$ if and only if $x = y$, otherwise $d_0(x,y) = 1$.

This chapter considers two families of pseudo-Boolean fitness landscapes specified in the next Table 11.2, using the pseudo-Boolean fitness functions stated earlier

in Section 11.2.1 over d_0-metric spaces and binary Hamming metric spaces. Hereinafter, complete graphs and finite d_0-metric spaces will be used interchangeably when convenient to denote or refer to the structure of those landscapes; the same applies to binary Hamming graphs and binary Hamming metric spaces.

Fitness landscape (family 1)	Fitness landscape (family 2)
$(\{0,1\}^n, d_0, f_{\text{OneMax}})$	$(\{0,1\}^n, d_{\text{H}}, f_{\text{OneMax}})$
$(\{0,1\}^n, d_0, f_{\text{Needle}})$	$(\{0,1\}^n, d_{\text{H}}, f_{\text{Needle}})$
$(\{0,1\}^n, d_0, f_{\text{LeadingOnes}})$	$(\{0,1\}^n, d_{\text{H}}, f_{\text{LeadingOnes}})$
$(\{0,1\}^n, d_0, f_{\text{NAE3SATP}})$	$(\{0,1\}^n, d_{\text{H}}, f_{\text{NAE3SATP}})$
$(\{0,1\}^n, d_0, f_{\text{WPP}})$	$(\{0,1\}^n, d_{\text{H}}, f_{\text{WPP}})$

Table 11.2. Two families of pseudo-Boolean fitness landscapes defined on d_0-metric spaces and binary Hamming metric spaces respectively.

11.3 Pseudo-Boolean Landscapes Classification

This section classifies the two families of pseudo-Boolean fitness landscapes proposed in Table 11.2 in Section 11.2. Table 11.3 below summarises the classification resulting from the rest of this section. The details of the classes of abstract convex landscapes and elementary landscapes classes are covered in Chapters 9 and 10.

Class	$(\{0,1\}^n, d_0)$	$(\{0,1\}^n, d_{\text{H}})$
Convex	All	—
Concave	All	—
Quasi-convex	All	—
Quasi-concave	All	Needle, leading-ones
Average-convex	All	One-max
Average-concave	All	One-max
(Tight) Fujiyama EL (order $p = 1$)	All	One-max
Non-Fujiyama EL (order $p > 1$)	—	NAE3SATP, WPP
Not elementary	—	Needle, leading-ones

(—) None of: one-max, needle, leading-ones, NAE3SATP and WPP.
(All) All of: one-max, needle, leading-ones, NAE3SATP and WPP.

Table 11.3. Classification of pseudo-Boolean fitness functions based on d_0-metric spaces $(\{0,1\}^n, d_0)$ and binary Hamming metric spaces $(\{0,1\}^n, d_{\text{H}})$ with respect to classes of abstract convex landscapes and elementary landscapes (ELs).

11.3.1 Classification in Discrete-distance Metric Spaces

Stadler [137] proves that for every real-valued non-constant function f defined on an arbitrary complete graph there exists some offset $c \in \mathbb{R}$ that makes $f - c\mathbf{1}$ elementary. Moreover, every elementary landscape on complete graphs is tight in the sense of Definition 10.8 according to Tlusty [145]. From these two facts and Lemma 11.1, it follows immediately that all non-flat fitness landscapes defined on complete graphs or finite d_0-metric spaces are tight Fujiyama elementary landscapes, that is Theorem 11.1.

Lemma 11.1. For every d_0-metric space (X, d_0), the identity crossover Id and the d_0-metric segment between every two parents $x, y \in X$ fulfil the relation $\text{Id}(x, y) = \{x, y\} = [x, y]_{d_0}$.

Proof. Fix two arbitrary parents $x, y \in X$. Either $x = y$, so $d_0(x, y) = 0$, or $x \neq y$, so $d_0(x, y) = 1$. If $x = y$, then $\text{Id}(x, x) = \{x\}$ by Definition 4.4, and $[x, x]_{d_0} = \{x\}$ by Definition 3.2 of metric segments. If $x \neq y$, then $\text{Id}(x, y) = \{x, y\}$ and $[x, y]_{d_0} = \{z \mid d_0(x, z) + d_0(z, y) = 1\} = \{x, y\}$ again by Definitions 4.4 and 3.2 respectively. Notice when $z \neq x$ and $z \neq y$, $d_0(x, z) + d_0(z, y) = 2 > 1$; therefore, $z = x$ or $z = y$ must hold so that: $d_0(x, z) = 0$ and $d_0(z, y) = 1$, if $z = x$, or $d_0(z, y) = 0$ and $d_0(x, z) = 1$ if $z = y$. Hence $\text{Id}(x, y) = \{x, y\} = [x, y]_{d_0}$. □

Theorem 11.1. Let $f : X \to \mathbb{R}$ be an arbitrary non-constant function defined on any d_0-metric space (X, d_0) where $|X| = n$ is finite. Then, there exists $c \in \mathbb{R}$ such that $f - c\mathbf{1}$ is a graph Laplacian eigenfunction with eigenvalue $\lambda_p = n$ of order $p = 1$ for the complete graph K_n. That is, $(X, d_0, f - c\mathbf{1})$ is a tight Fujiyama elementary landscape.

Proof. Lemma 11.1 proves $[x, y]_{d_0} = \{x, y\}$ for all $x, y \in X$. By Definition 11.6, $d_0(x, y) = 1$ if $x \neq y$, and $d_0(x, y) = 0$ if $x = y$. Given $|X| = n$, then (X, d_0) must be isomorphic to a complete graph K_n with $Spectrum(K_n) = \{0_{[1]}, n_{[n-1]}\}$ [13]. According to Remark 10.1, (X, d_0, f) is elementary if and only if there is an offset $c \in \mathbb{R}$ such that $f - c\mathbf{1}$ is a K_n graph Laplacian eigenfunction. Due to Stadler's Lemma 4 [137], any non-constant function on complete graphs is a graph Laplacian eigenfunction for some offset c. Moreover, any non-constant eigenfunction is orthogonal to the order-zero constant eigenfunction with eigenvalue $\lambda_0 = 0$ by Definition 10.2. Therefore, $f - c\mathbf{1}$ is a K_n graph Laplacian eigenfunction with eigenvalue $\lambda_p = n$ of order $p = 1$, hence (X, d_0, f) is a Fujiyama elementary landscape and tight as in Definition 10.8 due to Proposition 10.7 (see Tlusty's Corollary 2.6 [145]). □

Corollary 11.1 (To Theorem 11.1)**.** Let f be any of the one-max, needle-in-a-haystack, leading-ones, NAE3SATP, or WPP pseudo-Boolean fitness functions. Then, there exists an offset $c \in \mathbb{R}$ such that $(\{0,1\}^n, d_0, f - c\mathbf{1})$ is a tight Fujiyama elementary landscape.

Proof. It follows from Theorem 11.1. □

The next Theorem 11.2 and Corollary 11.2 prove that all fitness landscapes defined on complete graphs or d_0-metric spaces belong simultaneously to all these classes: convex, concave, quasi-convex, quasi-concave, average-convex and average-concave.

Theorem 11.2. Every function $f : X \to \mathbb{R}$ defined on an arbitrary d_0-metric space (X, d_0) is d_0-metric convex and d_0-metric concave. That is, (X, d_0, f) is a convex and concave fitness landscape.

Proof. According to Remark 9.4, it suffices to prove (X, d_0, f) is affine to prove (X, d_0, f) is convex and concave. That is, to verify

$$f(z) = \frac{d_0(y,z)}{d_0(x,y)} f(x) + \frac{d_0(x,z)}{d_0(x,y)} f(y) \tag{11.1}$$

holds $\forall x, y \in X$, such that $d_0(x,y) \neq 0$, and $\forall z \in [x,y]_{d_0}$. From Lemma 11.1, it follows that either $z = x$ or $z = y$ since $\forall x, y \in X : [x,y]_{d_0} = \{x, y\}$. If $z = x$, then substituting in Equation 11.1 yields

$$f(x) = \frac{1}{1} f(x) + \frac{0}{1} f(y) = f(x) \ ;$$

and, if $z = y$, then substituting in Equation 11.1 yields

$$f(y) = \frac{0}{1} f(x) + \frac{1}{1} f(y) = f(y) \ .$$

Therefore, f is d_0-metric convex and d_0-metric concave because it is affine. □

Corollary 11.2 (To Theorem 11.2)**.** Every function $f : X \to \mathbb{R}$ defined on an arbitrary d_0-metric space (X, d_0) is d_0-metric quasi-convex, d_0-metric quasi-concave, d_0-metric average-convex, and d_0-metric average-concave.

Proof. Let us begin with two observations:

(a) Theorem 11.2 proves every (X, d_0, f) is a convex and concave fitness landscape, where f is a d_0-metric convex and concave function.

(b) Lemma 11.1 proves $[x, y]_{d_0} = \{x, y\}$ for all $x, y \in X$. Hence d_0-metric spaces (X, d_0) have regular metric segments: $|\{z : z \in [x, y]_{d_0} \wedge d_0(x, z) = t\}| = |\{x, y\}| = |\{z : z \in [x, y]_{d_0} \wedge d_0(y, z) = t\}|$ for all $t \in \{0, \ldots, d_0(x, y)\}$.

From (a) it follows that (X, d_0, f) fulfils Definition 9.3 of quasi-convex and quasi-concave fitness landscapes due to Proposition 9.3. From both (a) and (b) it follows that (X, d_0, f) fulfils Definition 9.6 of average-convex and average-concave fitness landscapes due to Proposition 9.1. □

Corollary 11.3 (To Theorem 11.2 and Corollary 11.2)**.** Let f be any of the one-max, needle-in-a-haystack, leading-ones, NAE3SATP, or WPP pseudo-Boolean fitness functions. Then, $(\{0, 1\}^n, d_0, f)$ is convex, concave, quasi-convex, quasi-concave, average-convex, average-concave.

Proof. It follows directly from Theorem 11.2 and Corollary 11.2. □

11.3.2 Classification in Binary Hamming Metric Spaces

Preliminary research [48] suggests that, on binary Hamming graphs, only Fujiyama elementary landscapes (i.e. order-one graph Laplacian eigenfunctions) could belong to the convex, quasi-convex or average-convex classes or their concave versions. This section proves to what extent that is true for the second family of pseudo-Boolean fitness landscapes proposed earlier in Table 11.2.

One-max

Langdon [87], Chicano and others [23] show that the zero-averaged one-max fitness function $f_{\textsc{OneMax}} - \bar{f}_{\textsc{OneMax}}$, where $\bar{f}_{\textsc{OneMax}} = \frac{n}{2}$ is the global average, is a sum of order $p = 1$ Walsh functions; each of which is a graph Laplacian eigenfunction with eigenvalue $\lambda_p = 2p = 2$ for an n-dimensional binary Hamming graph (see Equation 10.8). That is, $(\{0, 1\}^n, d_{\mathrm{H}}, f_{\textsc{OneMax}})$ are Fujiyama elementary landscapes and tight due to Proposition 10.7. In addition, one-max fitness landscapes $(\{0, 1\}^n, d_{\mathrm{H}}, f_{\textsc{OneMax}})$ are:

- Average-convex and average-concave by Theorem 11.3, which relies on Proposition 11.1 proved by Chicano and others in Theorem 2 [23], taking into account that unbiased uniform crossover is a geometric crossover with offspring uniformly distributed over the Hamming metric segment between parents [100].
- Not quasi-convex nor quasi-concave by Theorem 11.4.

- Not convex nor concave by Corollary 11.4.

Proposition 11.1 (**One-max: expected offspring fitness** [23])**.** Let $(\{0,1\}^n, d_{\mathrm{H}}, f_{\textsc{OneMax}})$ be a one-max fitness landscape. Then, for all parents $x, y \in \{0,1\}^n$ and offspring $z \sim \mathrm{Unif}([x,y]_{d_{\mathrm{H}}})$:

$$\mathbb{E}[f_{\textsc{OneMax}}(z)] = |x \wedge y| + \frac{1}{2}|x \wedge \bar{y}| + \frac{1}{2}|\bar{x} \wedge y| \quad , \tag{11.2}$$

where $|x|$ is the number of `1` bits of x, $\bar{x}$ is the bitwise negation (NOT) of x, and $x \wedge y$ is the bitwise conjunction (AND) of x, y.

Theorem 11.3. One-max fitness landscapes $(\{0,1\}^n,\ d_{\mathrm{H}}, f_{\textsc{OneMax}})$ are average-convex and average-concave. That is, for all parents $x, y \in \{0,1\}^n$ and offspring $z \sim \mathrm{Unif}([x,y]_{d_{\mathrm{H}}})$: $\mathbb{E}[f_{\textsc{OneMax}}(z)] = \frac{1}{2}\Big(f_{\textsc{OneMax}}(x) + f_{\textsc{OneMax}}(y)\Big)$.

Proof. Since $\mathbb{E}[f_{\textsc{OneMax}}(z)] = |x \wedge y| + \frac{1}{2}|x \wedge \bar{y}| + \frac{1}{2}|\bar{x} \wedge y|$ due to Proposition 11.1, it suffices to prove

$$|x \wedge y| + \frac{1}{2}|x \wedge \bar{y}| + \frac{1}{2}|\bar{x} \wedge y| = \frac{1}{2}\Big(f_{\textsc{OneMax}}(x) + f_{\textsc{OneMax}}(y)\Big) \quad . \tag{11.3}$$

Observe that $|x \oplus y| = |x \wedge \bar{y}| + |\bar{x} \wedge y|$, where $x \oplus y = (x \wedge \bar{y}) \vee (\bar{x} \wedge y)$ is the bitwise exclusive-or (XOR) of x and y, and $f_{\textsc{OneMax}}(x) = \sum_{i=1}^{n} x_i = |x|$ counts the number of `1` bits in x. Then, Equation 11.3 becomes Equation 11.4:

$$\begin{aligned} |x \wedge y| + \frac{1}{2}|x \oplus y| &= \frac{1}{2}(|x| + |y|) \quad , \\ |x \oplus y| &= |x| + |y| - 2|x \wedge y| \quad . \end{aligned} \tag{11.4}$$

By definition of XOR: $x_i \oplus y_i = 0 \iff x_i = y_i$, and $x_i \oplus y_i = 1 \iff x_i \neq y_i$. Hence $|x \oplus y|$ equals the Hamming distance $d_{\mathrm{H}}(x,y)$ between x and y. It is easy to notice $d_{\mathrm{H}}(x,y) = |x| + |y| - 2|x \wedge y|$ holds. If we count the number of `1` bits $|x|$ in x and those $|y|$ of y, namely $|x| + |y|$, we count once the bits where x and y differ but twice the bits shared by both. Subtracting twice the number of shared bits $2|x \wedge y|$ from $|x| + |y|$ yields $d_{\mathrm{H}}(x,y)$. Therefore, Equation 11.3 holds and so does $\mathbb{E}[f_{\textsc{OneMax}}(z)] = \frac{1}{2}\Big(f_{\textsc{OneMax}}(x) + f_{\textsc{OneMax}}(y)\Big)$. □

Theorem 11.4. One-max fitness landscapes $(\{0,1\}^n, d_{\mathrm{H}}, f_{\textsc{OneMax}})$ are not quasi-convex nor quasi-concave.

Proof. Not all the sub-level sets of $f_{\textsc{OneMax}}$ are geodesically convex with respect to

$(\{0,1\}^n, d_{\mathrm{H}})$. Consider

$$\begin{aligned} L_{\leq 1}(f_{\textsc{OneMax}}) &= \{x \in \{0,1\}^n \mid f_{\textsc{OneMax}}(x) \leq 1\} \\ &= \{(0,\ldots,0),(1,0,\ldots,0),(0,1,0,\ldots,0),\ldots(0,0,\ldots,0,1)\} \ , \end{aligned}$$

comprising the n-dimensional all-zeros sequence alongside the unit vectors. Since

$$[(1,0,\ldots,0),(0,\ldots,0,1)]_{d_{\mathrm{H}}} \ni (1,0,\ldots,0,1) \notin L_{\leq 1}(f_{\textsc{OneMax}}) \ ,$$

it follows that $\forall x, y \in L_{\leq 1}(f_{\textsc{OneMax}}) : [x,y]_{d_{\mathrm{H}}} \not\subseteq L_{\leq 1}(f_{\textsc{OneMax}})$. Therefore, $f_{\textsc{OneMax}}$ is not d_{H}-metric quasi-convex, and so $(\{0,1\}^n, d_{\mathrm{H}}, f_{\textsc{OneMax}})$ is not quasi-convex by Definition 9.5. Likewise, not all the super-level sets of $f_{\textsc{OneMax}}$ are geodesically convex with respect to $(\{0,1\}^n, d_{\mathrm{H}})$. Consider

$$\begin{aligned} L_{\geq (n-1)}(f_{\textsc{OneMax}}) &= \{x \in \{0,1\}^n \mid f_{\textsc{OneMax}}(x) \geq n-1\} \\ &= \{(1,\ldots,1),(0,1,1,\ldots,1),(1,0,1,1,\ldots,1),\ldots,(1,\ldots,1,1,0)\} \ , \end{aligned}$$

comprising the n-dimensional binary sequences diametrically opposite to those of $L_{\leq 1}(f_{\textsc{OneMax}})$. Since

$$[(0,1,1,\ldots,1),(1,\ldots,1,1,0)]_{d_{\mathrm{H}}} \ni (0,1,\ldots,1,0) \notin L_{\geq (n-1)}(f_{\textsc{OneMax}}) \ ,$$

it follows that $\forall x, y \in L_{\geq (n-1)}(f_{\textsc{OneMax}}) : [x,y]_{d_{\mathrm{H}}} \not\subseteq L_{\geq (n-1)}(f_{\textsc{OneMax}})$. Therefore, $f_{\textsc{OneMax}}$ is not d_{H}-metric quasi-concave, and so $(\{0,1\}^n, d_{\mathrm{H}}, f_{\textsc{OneMax}})$ is not quasi-concave by Definition 9.5 and Remark 9.6. □

Corollary 11.4 (To Theorem 11.4)**.** One-max fitness landscapes $(\{0,1\}^n, d_{\mathrm{H}}, f_{\textsc{OneMax}})$ are not convex nor concave.

Proof. Due to Proposition 9.3, all convex fitness landscapes are quasi-convex, and all concave fitness landscapes are quasi-concave. But Theorem 11.4 proves $(\{0,1\}^n, d_{\mathrm{H}}, f_{\textsc{OneMax}})$ is not quasi-convex nor quasi-concave. Therefore, $(\{0,1\}^n, d_{\mathrm{H}}, f_{\textsc{OneMax}})$ is not convex nor concave. □

Needle-in-a-haystack

Langdon [87] proves that the fitness function of the needle-in-a-haystack problem induces an elementary landscape only when defined on complete graphs, implying it would not be elementary for instance on binary Hamming graphs. This section shows that needle fitness landscapes $(\{0,1\}^n, d_{\mathrm{H}}, f_{\textsc{Needle}})$ are:

- Not elementary landscapes by Theorem 11.5, which provides an alternative proof to Langdon's for binary Hamming graphs.
- Quasi-concave but not quasi-convex by Theorem 11.6.
- Not convex nor concave by Corollary 11.5 and Theorem 11.7 respectively.
- Not average-convex nor average-concave by Theorem 11.8.

Theorem 11.5. Needle fitness landscapes $(\{0,1\}^n, d_{\mathrm{H}}, f_{\textsc{Needle}})$ are not elementary.

Proof. According to Remark 10.1, if $(\{0,1\}^n, d_{\mathrm{H}}, f_{\textsc{Needle}})$ was elementary, there would be an offset $c \in \mathbb{R}$ such that $f'_{\textsc{Needle}} = f_{\textsc{Needle}} - c\mathbf{1}$ is an eigenfunction of the graph Laplacian for the n-dimensional binary Hamming graph $H(n,2)$ induced by $(\{0,1\}^n, d_{\mathrm{H}})$. Let us prove no such offset can exist.

From Definition 11.2, the image of $f_{\textsc{Needle}}$ is always 0 except for the all-ones binary sequence $(1,1,\ldots,1)$ that has image 1. So $f_{\textsc{Needle}}$ has the vector form $f_{\textsc{Needle}} = (f_{\textsc{Needle}}(x_0), f_{\textsc{Needle}}(x_1), \ldots, f_{\textsc{Needle}}(x_{2^n-2}), f_{\textsc{Needle}}(x_{2^n-1})) = (0,0,\ldots,0,1)$, where x_i, $0 \leq i \leq 2^n - 1$, is the binary sequence whose decimal number representation is i. Clearly, $f_{\textsc{Needle}}$ is non-constant. Any non-constant graph Laplacian eigenfunction must be orthogonal to the order-zero constant eigenfunction, thus $f'_{\textsc{Needle}}$ should have a non-zero eigenvalue and satisfy for some offset c

$$\langle f'_{\textsc{Needle}}, \mathbf{1} \rangle = \sum_{x \in \{0,1\}^n} f'_{\textsc{Needle}}(x) = \sum_{x \in \{0,1\}^n} (f_{\textsc{Needle}}(x) - c) = 0 \tag{11.5}$$

to be elementary. Only one c satisfies Equation 11.5:

$$\begin{aligned} \sum_{x \in \{0,1\}^n} (f_{\textsc{Needle}}(x) - c) &= 0 \\ \underbrace{(-c) + (-c) + \cdots + (-c)}_{2^{n-1} \text{ times}} + (1 - c) &= 0 \\ -2^n \cdot c + 1 &= 0 \\ c &= \frac{1}{2^n} \ . \end{aligned}$$

Precisely, $c = \frac{1}{2^n} = \bar{f}_{\textsc{Needle}}$ is the global average of $f_{\textsc{Needle}}$. Hence $f'_{\textsc{Needle}} = f_{\textsc{Needle}} - \bar{f}\mathbf{1} = (-\frac{1}{2^n}, -\frac{1}{2^n}, \ldots, -\frac{1}{2^n}, \frac{2^n-1}{2^n})$. However, it is plain to see $f'_{\textsc{Needle}}$ is not a $H(n,2)$ graph Laplacian eigenfunction: no eigenvalue $\lambda > 0$ would satisfy $(\mathbf{L}f'_{\textsc{Needle}})(x_i) = \lambda \cdot f'_{\textsc{Needle}}(x_i)$ for all x_i. For example, one may check x_0 already yields $(\mathbf{L}f'_{\textsc{Needle}})(x_0) = 0$ using Equation 10.3 for the single bit-flip neighbourhood

function BITPLIP(1), Definition 4.1, associated with $H(n,2)$:

$$\begin{aligned}(\mathbf{L}f'_{\text{NEEDLE}})(x_0) &= |\text{BITPLIP}(1)(x_0)| \cdot f'_{\text{NEEDLE}}(x_0) - \sum_{y \in \text{BITPLIP}(1)(x_0)} f'_{\text{NEEDLE}}(y) \\ &= n \cdot \left(-\frac{1}{2^n}\right) - \underbrace{\left(\left(-\frac{1}{2^n}\right) + \cdots + \left(-\frac{1}{2^n}\right)\right)}_{n \text{ times due to } n \text{ neighbours}} \\ &= -\frac{n}{2^n} + \frac{n}{2^n} = 0 \ .\end{aligned}$$

That is, $(\mathbf{L}f'_{\text{NEEDLE}})(x_0) = 0 \neq \lambda \cdot (-\frac{1}{2^n}) = \lambda \cdot f'_{\text{NEEDLE}}(x_0)$ for all $\lambda > 0$. □

Theorem 11.6. Needle fitness landscapes $(\{0,1\}^n, d_{\text{H}}, f_{\text{NEEDLE}})$ are quasi-concave, but they are not quasi-convex.

Proof. From Definition 11.2, the image of f_{NEEDLE} is always 0 except for the all-ones binary sequence that has image 1. So f_{NEEDLE} has two super-level sets

$$\begin{aligned}L_{\geq 0}(f_{\text{NEEDLE}}) &= \{0,1\}^n && \equiv \texttt{**}\cdots\texttt{*} \ , \\ L_{\geq 1}(f_{\text{NEEDLE}}) &= \{1\}^n && \equiv \texttt{11}\cdots\texttt{1} \ ,\end{aligned}$$

and two sub-level sets

$$\begin{aligned}L_{\leq 0}(f_{\text{NEEDLE}}) &= \{0,1\}^n \setminus \{1\}^n \ , \\ L_{\leq 1}(f_{\text{NEEDLE}}) &= \{0,1\}^n && \equiv \texttt{**}\cdots\texttt{*} \ .\end{aligned}$$

The super-level sets are schemas or invariant subsets (see Section 6.2 in Chapter 6), which are geodesically convex sets for $(\{0,1\}^n, d_{\text{H}})$ as a consequence of Proposition A.1 by Mitavskiy [98]. Therefore, f_{NEEDLE} is a d_{H}-metric quasi-concave function, so $(\{0,1\}^n, d_{\text{H}}, f_{\text{NEEDLE}})$ is quasi-concave according to Definition 9.5 and Remark 9.6. However, the sub-level set $L_{\leq 0}(f_{\text{NEEDLE}})$ is not geodesically convex since

$$[(1,0,1,0\ldots,1,0),(0,1,0,1\ldots,0,1)]_{d_{\text{H}}} \ni (1,\ldots,1) \notin L_{\leq 0}(f_{\text{NEEDLE}}) \ ,$$

that is $\forall x, y \in L_{\leq 0}(f_{\text{NEEDLE}}) : [x,y]_{d_{\text{H}}} \not\subseteq L_{\leq 0}(f_{\text{NEEDLE}})$. Therefore, f_{NEEDLE} is not a d_{H}-metric quasi-convex function, and so $(\{0,1\}^n, d_{\text{H}}, f_{\text{NEEDLE}})$ is not quasi-convex according to Definition 9.5. □

Corollary 11.5 (To Theorem 11.6)**.** Needle fitness landscapes $(\{0,1\}^n, d_{\text{H}}, f_{\text{NEEDLE}})$ are not convex.

Proof. Every convex fitness landscape is quasi-convex due to Proposition 9.3, but Theorem 11.6 proves $(\{0,1\}^n, d_{\mathrm{H}}, f_{\textsc{Needle}})$ is not quasi-convex. Therefore, $(\{0,1\}^n, d_{\mathrm{H}}, f_{\textsc{Needle}})$ is not convex. □

Theorem 11.7. Needle fitness landscapes $(\{0,1\}^n, d_{\mathrm{H}}, f_{\textsc{Needle}})$ are not concave.

Proof. The following counterexample proves $(\{0,1\}^4, d_{\mathrm{H}}, f_{\textsc{Needle}})$ is not concave (see Definition 9.3). Fix $n = 4$, two parent bit strings $x = \texttt{0101}$ and $y = \texttt{1111}$, and one offspring $z = \texttt{0111} \in [x,y]_{d_{\mathrm{H}}}$. Then,

$$0 = f_{\textsc{Needle}}(z) < \frac{d_{\mathrm{H}}(y,z)}{d_{\mathrm{H}}(x,y)} f_{\textsc{Needle}}(x) + \frac{d_{\mathrm{H}}(x,z)}{d_{\mathrm{H}}(x,y)} f_{\textsc{Needle}}(y) = \frac{1}{2}0 + \frac{1}{2}1 = \frac{1}{2} \quad .$$

Therefore, needle fitness landscapes $(\{0,1\}^n, d_{\mathrm{H}}, f_{\textsc{Needle}})$ are not always concave. □

Theorem 11.8. Needle fitness landscapes $(\{0,1\}^n, d_{\mathrm{H}}, f_{\textsc{Needle}})$ are not average-convex nor average-concave.

Proof. The following two counterexamples prove $(\{0,1\}^4, d_{\mathrm{H}}, f_{\textsc{Needle}})$ are not average-convex nor average-concave respectively (see Definition 9.6). For the first counterexample, consider $n = 4$, two parent bit strings $x = \texttt{1101}$ and $y = \texttt{1110}$, and one offspring random variable $z \sim \mathrm{Unif}([x,y]_{d_{\mathrm{H}}}) = \mathrm{Unif}(\{\texttt{1100}, \texttt{1101}, \texttt{1110}, \texttt{1111}\})$. Then,

$$\mathbb{E}[f_{\textsc{Needle}}(z)] = \frac{1}{|[x,y]_{d_{\mathrm{H}}}|} \sum_{z \in [x,y]_{d_{\mathrm{H}}}} f_{\textsc{Needle}}(z) = \frac{1}{4} > 0 = \frac{f_{\textsc{Needle}}(x) + f_{\textsc{Needle}}(y)}{2} \quad .$$

Therefore, $(\{0,1\}^4, d_{\mathrm{H}}, f_{\textsc{Needle}})$ is not average-convex. For the second counterexample, consider two parent bit strings $x = \texttt{0000}$ and $y = \texttt{1111}$, and one offspring random variable $z \sim \mathrm{Unif}([x,y]_{d_{\mathrm{H}}}) = \mathrm{Unif}(\{0,1\}^4)$. Then,

$$\mathbb{E}[f_{\textsc{Needle}}(z)] = \frac{1}{|[x,y]_{d_{\mathrm{H}}}|} \sum_{z \in [x,y]_{d_{\mathrm{H}}}} f_{\textsc{Needle}}(z) = \frac{1}{16} < \frac{1}{2} = \frac{f_{\textsc{Needle}}(x) + f_{\textsc{Needle}}(y)}{2} \quad .$$

Therefore, $(\{0,1\}^4, d_{\mathrm{H}}, f_{\textsc{Needle}})$ is not average-concave. Hence needle fitness landscapes $(\{0,1\}^n, d_{\mathrm{H}}, f_{\textsc{Needle}})$ are not always average-convex or average-concave. □

Leading-ones

Leading-ones fitness landscapes $(\{0,1\}^n, d_{\mathrm{H}}, f_{\textsc{LeadingOnes}})$ are:

- Not elementary landscapes by Theorem 11.9.

- Quasi-concave but not quasi-convex by Theorem 11.10.
- Not convex nor concave Corollary 11.6 and Theorem 11.11 respectively.
- Not average-convex nor average-concave by Theorem 11.12.

Theorem 11.9. Leading-ones fitness landscapes $(\{0,1\}^n,\, d_{\mathrm{H}}, f_{\textsc{LeadingOnes}})$ are not elementary.

Proof. It suffices to note from its Definition 11.3 that

$$f_{\textsc{LeadingOnes}}(x) = \sum_{i=1}^{n} \prod_{j=1}^{i} x_j = x_1 + (x_1x_2) + (x_1x_2x_3) + \cdots + (x_1x_2x_3\cdots x_n)$$

is a sum of Walsh basis functions (Equation 10.7) through all orders $p \in \{1,\ldots,n\}$, each of which is a graph Laplacian eigenfunction corresponding to a different eigenvalue $\lambda_p = 2p$ (Equation 10.8) for the binary Hamming graph induced by $(\{0,1\}^n, d_{\mathrm{H}})$. Therefore, $(\{0,1\}^n, d_{\mathrm{H}}, f_{\textsc{LeadingOnes}})$ is not elementary. □

Theorem 11.10. Leading-ones fitness landscapes $(\{0,1\}^n, d_{\mathrm{H}}, f_{\textsc{LeadingOnes}})$ are quasi-concave, but they are not quasi-convex.

Proof. The super-level sets of $f_{\textsc{LeadingOnes}}$

$$\begin{aligned}
L_{\geq 0}(f_{\textsc{LeadingOnes}}) &= \{0,1\}^n && \equiv \texttt{***}\cdots\texttt{**}\ , \\
L_{\geq 1}(f_{\textsc{LeadingOnes}}) &= \{1\} \times \{0,1\}^{n-1} && \equiv \texttt{1**}\cdots\texttt{**}\ , \\
L_{\geq 2}(f_{\textsc{LeadingOnes}}) &= \{1\}^2 \times \{0,1\}^{n-2} && \equiv \texttt{11*}\cdots\texttt{**}\ , \\
&\vdots && \vdots \\
L_{\geq n-1}(f_{\textsc{LeadingOnes}}) &= \{1\}^{n-1} \times \{0,1\} && \equiv \texttt{111}\cdots\texttt{1*}\ , \\
L_{\geq n}(f_{\textsc{LeadingOnes}}) &= \{1\}^n && \equiv \texttt{111}\cdots\texttt{11}
\end{aligned}$$

are schemas or invariant subsets (see Section 6.2 in Chapter 6), which are geodesically convex sets for $(\{0,1\}^n, d_{\mathrm{H}})$ as a consequence of Proposition A.1 by Mitavskiy [98]. Therefore, $f_{\textsc{LeadingOnes}}$ is a d_{H}-metric quasi-concave function, so $(\{0,1\}^n, d_{\mathrm{H}}, f_{\textsc{LeadingOnes}})$ is quasi-concave according to Definitions 9.5–9.5 and Remark 9.6.

However, $(\{0,1\}^n, d_{\mathrm{H}}, f_{\textsc{LeadingOnes}})$ is not quasi-convex because not all the sub-level sets are geodesically convex. Consider, for example, $L_{\leq n-1}(f_{\textsc{LeadingOnes}}) = \{0,1\}^n \setminus \{1\}^n$. Then, $\forall x, y \in L_{\leq n-1}(f_{\textsc{LeadingOnes}}) : [x,y]_{d_{\mathrm{H}}} \not\subseteq L_{\leq n-1}(f_{\textsc{LeadingOnes}})$ since $[(1,0,1,0\ldots,1,0),(0,1,0,1\ldots,0,1)]_{d_{\mathrm{H}}} \ni (1,\ldots,1) \notin L_{\leq n-1}(f_{\textsc{LeadingOnes}})$. □

Corollary 11.6 (To Theorem 11.10)**.** Leading-ones fitness landscapes $(\{0,1\}^n, d_{\mathrm{H}}, f_{\textsc{LeadingOnes}})$ are not convex.

Proof. Every convex fitness landscape is quasi-convex due to Proposition 9.3, but Theorem 11.10 proves $(\{0,1\}^n, d_{\mathrm{H}}, f_{\textsc{LeadingOnes}})$ is not quasi-convex. Therefore, $(\{0,1\}^n, d_{\mathrm{H}}, f_{\textsc{LeadingOnes}})$ is not convex. □

Theorem 11.11. Leading-ones fitness landscapes $(\{0,1\}^n, d_{\mathrm{H}}, f_{\textsc{LeadingOnes}})$ are not concave.

Proof. The following counterexample proves $(\{0,1\}^4, d_{\mathrm{H}}, f_{\textsc{LeadingOnes}})$ is not concave (see Definition 9.3). Fix $n = 4$, two parent bit strings $x = \texttt{0011}$ and $y = \texttt{1111}$, and one offspring $z = \texttt{0111} \in [x,y]_{d_{\mathrm{H}}}$. Then,

$$\begin{aligned} 0 = f_{\textsc{LeadingOnes}}(z) &< \frac{d_{\mathrm{H}}(y,z)}{d_{\mathrm{H}}(x,y)} f_{\textsc{LeadingOnes}}(x) + \frac{d_{\mathrm{H}}(x,z)}{d_{\mathrm{H}}(x,y)} f_{\textsc{LeadingOnes}}(y) \\ &= \frac{1}{2}\,0 + \frac{1}{2}\,4 = 2 \ . \end{aligned}$$

Therefore, leading-ones fitness landscapes $(\{0,1\}^n, d_{\mathrm{H}}, f_{\textsc{LeadingOnes}})$ are not always concave. □

Theorem 11.12. Leading-ones fitness landscapes $(\{0,1\}^n, d_{\mathrm{H}}, f_{\textsc{LeadingOnes}})$ are not average-convex nor average-concave.

Proof. The following two counterexamples prove $(\{0,1\}^3, d_{\mathrm{H}}, f_{\textsc{LeadingOnes}})$ are not average-convex nor average-concave (see Definition 9.6). For the first counterexample, consider $n = 3$, two parent bit strings $x = \texttt{010}$ and $y = \texttt{100}$, and one offspring random variable $z \sim \mathrm{Unif}([x,y]_{d_{\mathrm{H}}}) = \mathrm{Unif}(\{\texttt{000}, \texttt{010}, \texttt{100}, \texttt{110}\})$. Then,

$$\begin{aligned} \mathbb{E}[f_{\textsc{LeadingOnes}}(z)] &= \frac{1}{|[x,y]_{d_{\mathrm{H}}}|} \sum_{z \in [x,y]_{d_{\mathrm{H}}}} f_{\textsc{LeadingOnes}}(z) = \frac{3}{4} \\ &> \frac{1}{2} = \frac{f_{\textsc{LeadingOnes}}(x) + f_{\textsc{LeadingOnes}}(y)}{2} \ . \end{aligned}$$

Therefore, $(\{0,1\}^3, d_{\mathrm{H}}, f_{\textsc{LeadingOnes}})$ is not average-convex. For the second counterexample, consider two parent bit strings $x = \texttt{000}$ and $y = \texttt{111}$, and one offspring random variable $z \sim \mathrm{Unif}([x,y]_{d_{\mathrm{H}}}) = \mathrm{Unif}(\{0,1\}^3)$. Then,

$$\begin{aligned} \mathbb{E}[f_{\textsc{LeadingOnes}}(z)] &= \frac{1}{|[x,y]_{d_{\mathrm{H}}}|} \sum_{z \in [x,y]_{d_{\mathrm{H}}}} f_{\textsc{LeadingOnes}}(z) = \frac{7}{8} \\ &< \frac{3}{2} = \frac{f_{\textsc{LeadingOnes}}(x) + f_{\textsc{LeadingOnes}}(y)}{2} \ . \end{aligned}$$

Therefore, $(\{0,1\}^3, d_{\mathrm{H}}, f_{\textsc{LeadingOnes}})$ is not average-concave. Hence leading-ones fitness landscapes $(\{0,1\}^n, d_{\mathrm{H}}, f_{\textsc{LeadingOnes}})$ are not always average-convex or average-concave. □

Not-all-equal 3-satisfiability and Weight Partitioning

Grover [56] proves that $(\{0,1\}^n, d_{\mathrm{H}}, f_{\mathrm{NAE3SATP}})$ and $(\{0,1\}^n, d_{\mathrm{H}}, f_{\mathrm{WPP}})$ are non-Fujiyama elementary landscapes or, in other words, graph Laplacian eigenfunctions of order $p = 2$ whose corresponding eigenvalue is $\lambda_p = 2p = 4$ as Klemm and Stadler [85] indicate.

Proposition 11.2 (**NAE3SATP, WPP: Non-Fujiyama EL** [56, 85])**.** Let $H(n,2)$ be the binary Hamming graph induced by $(\{0,1\}^n, d_{\mathrm{H}})$ for $n > 1$. Let f be any of the NAE3SATP or WPP pseudo-Boolean fitness functions. Then, there is an offset $c \in \mathbb{R}$ so that $f - c\mathbf{1}$ is a $H(n,2)$ graph Laplacian eigenfunction with eigenvalue $\lambda_p = 2p = 4$ of order $p = 2$. That is, $(\{0,1\}^n, d_{\mathrm{H}}, f)$ is a non-Fujiyama EL.

Additionally, $(\{0,1\}^n, d_{\mathrm{H}}, f_{\mathrm{NAE3SATP}})$ and $(\{0,1\}^n, d_{\mathrm{H}}, f_{\mathrm{WPP}})$ are:

- Not quasi-convex, quasi-concave, convex nor concave by Corollary 11.7, which follows from a more general Theorem 11.15 proved later in Section 11.4.1.
- Not average-convex nor average-concave by Theorems 11.13 and 11.14.

Corollary 11.7 (To Theorem 11.15)**.** Let $H(n,2)$ be the binary Hamming graph induced by $(\{0,1\}^n, d_{\mathrm{H}})$ for $n > 1$. Let f be any of the NAE3SATP or WPP pseudo-Boolean fitness functions. Then, $(\{0,1\}^n, d_{\mathrm{H}}, f)$ is not quasi-convex, quasi-concave, convex nor concave.

Proof. It follows directly from Theorem 11.15, and the fact that the f_{NAE3SATP} and f_{WPP} fitness functions are graph Laplacian eigenfunctions of order $p = 2$ for any binary Hamming graph $H(n,2)$ with $n > 1$ according to Proposition 11.2. □

Theorem 11.13. NAE3SATP fitness landscapes $(\{0,1\}^n, d_{\mathrm{H}}, f_{\mathrm{NAE3SATP}})$ are not average-convex nor average-concave.

Proof. It suffices to find an instance of NAE3SATP where $(\{0,1\}^n, d_{\mathrm{H}}, f_{\mathrm{NAE3SATP}})$ is not average-convex nor average-concave (see Definition 9.6). The following two counterexamples prove it so for the instance given by the family of clauses

$$\mathcal{C}_x = \{\{x_1, \bar{x}_2, \bar{x}_3\}, \{\bar{x}_1, x_3, \bar{x}_4\}, \{x_2, \bar{x}_3, x_4\}\}$$

parametrised by a binary variable sequence $x = (x_1, x_2, x_3, x_4) \in \{0,1\}^4$. For the first counterexample, let binary variable assignments be defined by the parent strings $u = (1,0,0,1) \equiv \texttt{1001}$ and $v = (1,0,1,0) \equiv \texttt{1010}$, where $w \sim \mathrm{Unif}([u,v]_{d_\mathrm{H}}) = \mathrm{Unif}(\{\texttt{1000}, \texttt{1001}, \texttt{1010}, \texttt{1011}\})$ is a uniformly random offspring. Then,

$$\begin{aligned}
f_{\mathrm{NAE3SATP}}(\texttt{1000}) &= 2 \quad \text{for} \quad \mathcal{C}_{\texttt{1000}} = \{\{1,1,1\},\{0,0,1\},\{0,1,0\}\}\,; \\
f_{\mathrm{NAE3SATP}}(\texttt{1001}) &= 1 \quad \text{for} \quad \mathcal{C}_{\texttt{1001}} = \{\{1,1,1\},\{0,0,0\},\{0,1,1\}\}\,; \\
f_{\mathrm{NAE3SATP}}(\texttt{1010}) &= 2 \quad \text{for} \quad \mathcal{C}_{\texttt{1010}} = \{\{1,1,0\},\{0,1,1\},\{0,0,0\}\}\,; \\
f_{\mathrm{NAE3SATP}}(\texttt{1011}) &= 3 \quad \text{for} \quad \mathcal{C}_{\texttt{1011}} = \{\{1,1,0\},\{0,1,0\},\{0,0,1\}\}\ .
\end{aligned}$$

Hence

$$\begin{aligned}
\mathbb{E}[f_{\mathrm{NAE3SATP}}(w)] &= \frac{1}{|[u,v]_{d_\mathrm{H}}|} \sum_{w \in [u,v]_{d_\mathrm{H}}} f_{\mathrm{NAE3SATP}}(w) = \frac{8}{4} = 2 \\
&> \frac{3}{2} = \frac{f_{\mathrm{NAE3SATP}}(u) + f_{\mathrm{NAE3SATP}}(v)}{2}\ .
\end{aligned}$$

Therefore, $(\{0,1\}^4, d_\mathrm{H}, f_{\mathrm{NAE3SATP}})$ is not average-convex for the instance $\mathcal{C}_x$. For the second counterexample, consider the same instance $\mathcal{C}_x$ but parents $u' = \texttt{1110}$ and $v' = \texttt{1011}$, such that $w' \sim \mathrm{Unif}([u',v']_{d_\mathrm{H}}) = \mathrm{Unif}(\{\texttt{1010}, \texttt{1011}, \texttt{1110}, \texttt{1111}\})$ is a uniformly random offspring. Then,

$$\begin{aligned}
f_{\mathrm{NAE3SATP}}(\texttt{1010}) &= 2 \quad \text{for} \quad \mathcal{C}_{\texttt{1010}} = \{\{1,1,0\},\{0,1,1\},\{0,0,0\}\}; \\
f_{\mathrm{NAE3SATP}}(\texttt{1011}) &= 3 \quad \text{for} \quad \mathcal{C}_{\texttt{1011}} = \{\{1,1,0\},\{0,1,0\},\{0,0,1\}\}; \\
f_{\mathrm{NAE3SATP}}(\texttt{1110}) &= 3 \quad \text{for} \quad \mathcal{C}_{\texttt{1110}} = \{\{1,0,0\},\{0,1,1\},\{1,0,0\}\}; \\
f_{\mathrm{NAE3SATP}}(\texttt{1111}) &= 3 \quad \text{for} \quad \mathcal{C}_{\texttt{1111}} = \{\{1,0,0\},\{0,1,0\},\{1,0,1\}\}\ .
\end{aligned}$$

Hence

$$\begin{aligned}
\mathbb{E}[f_{\mathrm{NAE3SATP}}(w')] &= \frac{1}{|[u',v']_{d_\mathrm{H}}|} \sum_{w' \in [u',v']_{d_\mathrm{H}}} f_{\mathrm{NAE3SATP}}(w') = \frac{11}{4} = 2.75 \\
&< 3 = \frac{6}{2} = \frac{f_{\mathrm{NAE3SATP}}(u') + f_{\mathrm{NAE3SATP}}(v')}{2}\ .
\end{aligned}$$

Therefore, $(\{0,1\}^4, d_\mathrm{H}, f_{\mathrm{NAE3SATP}})$ is not average-concave for the instance $\mathcal{C}_x$. Consequently, NAE3SATP fitness landscapes $(\{0,1\}^n, d_\mathrm{H}, f_{\mathrm{NAE3SATP}})$ are not always average-convex or average-concave. □

Theorem 11.14. WPP fitness landscapes $(\{0,1\}^n, d_\mathrm{H}, f_{\mathrm{WPP}})$ are not average-convex nor average-concave.

Proof. It suffices to find an instance of WPP where $(\{0,1\}^n, d_{\mathrm{H}}, f_{\mathrm{WPP}})$ is not average-convex nor average-concave (see Definition 9.6). The following two counterexamples prove it so for $n = 4$ on the instance given by the sequence of weights $w = (-6, 5, -2, 1)$. For the first counterexample, let two-way partitions on weights given by binary string parents $x = (1,0,0,1) \equiv \texttt{1001}$ and $y = (1,1,1,0) \equiv \texttt{1110}$, with a uniform random variable offspring $z \sim \mathrm{Unif}([x,y]_{d_{\mathrm{H}}}) = \mathrm{Unif}(\{1\} \times \{0,1\}^3)$. Then,

$$\begin{aligned} f_{\mathrm{WPP}}(\texttt{1000}) &= 100, & f_{\mathrm{WPP}}(\texttt{1100}) &= 0, \\ f_{\mathrm{WPP}}(\texttt{1001}) &= 64, & f_{\mathrm{WPP}}(\texttt{1101}) &= 4, \\ f_{\mathrm{WPP}}(\texttt{1010}) &= 196, & f_{\mathrm{WPP}}(\texttt{1110}) &= 16, \\ f_{\mathrm{WPP}}(\texttt{1011}) &= 144, & f_{\mathrm{WPP}}(\texttt{1111}) &= 4 \ . \end{aligned}$$

Hence

$$\begin{aligned} \mathbb{E}[f_{\mathrm{WPP}}(z)] &= \frac{1}{|[x,y]_{d_{\mathrm{H}}}|} \sum_{z \in [x,y]_{d_{\mathrm{H}}}} f_{\mathrm{WPP}}(z) = \frac{528}{8} = 66 \\ &> 40 = \frac{80}{2} = \frac{f_{\mathrm{WPP}}(x) + f_{\mathrm{WPP}}(y)}{2} \ . \end{aligned}$$

Therefore, $(\{0,1\}^4, d_{\mathrm{H}}, f_{\mathrm{WPP}})$ is not average-convex for the instance w. For the second counterexample, consider the same instance of weights w but parents $x' = \texttt{0000}$ and $y' = \texttt{1010}$, such that $z' \sim \mathrm{Unif}([x',y']_{d_{\mathrm{H}}}) = \mathrm{Unif}(\texttt{0000}, \texttt{0010}, \texttt{1000}, \texttt{1010})$ is a uniformly random offspring. Then,

$$f_{\mathrm{WPP}}(\texttt{0000}) = 4, \ f_{\mathrm{WPP}}(\texttt{0010}) = 4, \ f_{\mathrm{WPP}}(\texttt{1000}) = 100, \ f_{\mathrm{WPP}}(\texttt{1010}) = 196 \ .$$

Hence

$$\begin{aligned} \mathbb{E}[f_{\mathrm{WPP}}(z')] &= \frac{1}{|[x',y']_{d_{\mathrm{H}}}|} \sum_{z' \in [x',y']_{d_{\mathrm{H}}}} f_{\mathrm{WPP}}(z') = \frac{304}{4} = 76 \\ &< 100 = \frac{200}{2} = \frac{f_{\mathrm{WPP}}(x') + f_{\mathrm{WPP}}(y')}{2} \ . \end{aligned}$$

Therefore, $(\{0,1\}^4, d_{\mathrm{H}}, f_{\mathrm{WPP}})$ is not average-concave for the instance w. Consequently, WPP fitness landscapes $(\{0,1\}^n, d_{\mathrm{H}}, f_{\mathrm{WPP}})$ are not always average-convex or average-concave. □

11.4 Abstract Convex Elementary Landscapes

Proving the existence of abstract convex elementary landscapes to address Question 11.1 has been the focus of this chapter thus far. This section considers other aspects of abstract convex elementary landscapes not covered previously which also concern Questions 11.1 and 11.2. First, Section 11.4.1 shows that (quasi-)convex or (quasi-)concave elementary landscapes generally correspond to graph Laplacian eigenfunctions of order less than two. Then, Section 11.4.2 explains what characterises certain perturbed fitness landscapes as being elementary and approximately (average or quasi) convex.

11.4.1 Quasi-convexity and Convexity

Graph Laplacian eigenfunctions of order $p \geq 2$, that is non-Fujiyama elementary landscapes, are not weakly unimodal (i.e. may not satisfy Definition 10.8). By contrast, Fujiyama elementary landscapes (of order $p = 1$) are always tight (and thus always weakly unimodal, or weakly tight, by extension) for certain classes of graphs (see Section 10.4). This dichotomy separates quasi-convex and non-quasi-convex elementary landscapes, as well as convex and non-convex elementary landscapes, as implied by the next Theorem 11.15 since Lemma 11.2 proves quasi-convex functions are always weakly unimodal on connected graphs.

Lemma 11.2. Every d-metric quasi-convex, or quasi-concave, function defined on the vertex set of any finite connected graph, with shortest-path metric d, is weakly unimodal.

Proof. Consider a finite connected graph G with vertex set V and shortest-path metric d, where $f : V \to \mathbb{R}$ is a d-metric quasi-convex function. All sub-level sets are vertex subsets $L_{\leq \ell}(f) \subseteq V$ for all $\ell \in \mathbb{R}$; and, by Definition 9.5 of d-metric quasi-convex function, all such $L_{\leq \ell}(f)$ are geodesically convex: $[x, y]_d \subseteq L_{\leq \ell}(f)$ for all $x, y \in L_{\leq \ell}(f)$ and $\ell \in \mathbb{R}$. Therefore, each subgraph $G[L_{\leq \ell}(f)]$ induced on G by the vertex subset $L_{\leq \ell}(f)$ must be connected since $[x, y]_d = \{z \in V \mid d(x, z) + d(z, y) = d(x, y)\} \subseteq L_{\leq \ell}(f)$, $\forall x, y \in L_{\leq \ell}(f)$, implies: a (shortest) path exists between every pair of vertices $x, y \in L_{\leq \ell}(f)$, and all vertices z of that path lie in $L_{\leq \ell}(f)$. Hence, by Definition 10.8, f is weakly unimodal because all $G[L_{\leq \ell}(f)]$ are connected. The proof follows analogously for d-metric quasi-concave functions by showing that all super-level sets $L_{\geq \ell}(f)$ induce connected subgraphs $G[L_{\geq \ell}(f)]$. □

Remark 11.1 (To Lemma 11.2)**.** The notions of unimodality in Definition 10.8 and quasi-convexity in Definition 9.5 allow fitness landscapes with *plateaus* of arbitrary size (i.e. regions where fitness remains constant). For instance, any flat landscape would be regarded weakly unimodal and quasi-convex.

Theorem 11.15. Let f be any (real-valued) graph Laplacian eigenfunction corresponding to an eigenvalue $\lambda_p \geq 0$ for any finite connected graph G with shortest-path metric d. If f is quasi-convex, quasi-concave, convex or concave with respect to the metric d, then the eigenvalue λ_p is of order $p < 2$.

Proof. For a general graph Laplacian eigenfunction f of order $p \geq 2$ on G, the subgraphs $G[L_{\leq \ell}(f)]$ induced by all sub-level sets of f, and the subgraphs $G[L_{\geq \ell}(f)]$ induced by all super-level sets of f, need not be connected $\forall \ell \in \mathbb{R}$ [145]. That is, f need not be weakly unimodal as in Definition 10.8. But Lemma 11.2 proves that every d-metric quasi-convex or quasi-concave function on G must be weakly unimodal always. Therefore, f need not be d-metric quasi-convex nor quasi-concave. Due to Proposition 9.3 convex functions are quasi-convex, and concave functions are quasi-concave, so f is not convex nor concave with respect to the metric d. □

Corollary 11.8 (To Theorem 11.15)**.** If a non-constant graph Laplacian eigenfunction f is also (quasi-)convex or (quasi-)concave, for a given arbitrary finite connected graph, then f has exactly two weak discrete nodal domains.

Proof. It follows from Proposition 10.6 (see Corollary 3.2 in [8]), given that f corresponds to a graph Laplacian eigenvalue of order $p = 1$ (i.e. smallest non-zero eigenvalue) since $p < 2$ due to Theorem 11.15 and $0 < p$ because f is non-constant. □

Put differently, Theorem 11.15 says there are no non-flat quasi-convex elementary landscapes (nor convex ones by inclusion) corresponding to eigenvalues of order $p \geq 2$ in general. Nevertheless, it remains possible for Fujiyama elementary landscapes (i.e. order $p = 1$) to be quasi-convex or convex. In fact, we already saw in Section 11.3.1 that all non-flat quasi-convex or convex elementary landscapes on complete graphs are of order $p = 1$ (see Theorem 11.1 and Corollary 11.2).

The eigenvalue order upper-bound $p < 2$ in Theorem 11.15 becomes sharp $p = 0$ for convex (or concave) functions in the following case. There exist a class of graphs, which includes all non-trivial binary Hamming graphs, where every convex function (with respect to the shortest-path metric) is also a sub-harmonic function [16]. We prove next that only flat elementary landscapes, namely order-zero graph Laplacian eigenfunctions, can be sub-harmonic; therefore, only flat elementary landscapes can be convex on the aforesaid class of graphs.

Convex and Sub-harmonic Elementary Landscapes

Moraglio [101] observes that convex or concave fitness functions, with respect to the Hamming metric, are flat when defined on binary Hamming graphs. Notably, n-dimensional binary Hamming graphs $H(n, 2)$ for $n > 1$ belong to a larger family of finite connected graphs, that is triangle-free graphs with vertices of degree two or higher [14]; in other words, graphs without length-three cycles (i.e. isomorphic copies of complete graphs K_3) whose vertices have two or more adjacent neighbours. Theorem 9 by Burke and Perkins [16] (see Proposition 11.3) proves that for such family of graphs all convex functions are *sub-harmonic functions* (Definition 11.7). The next Lemma 11.3 proves that only flat elementary landscapes can be sub-harmonic regardless of the underlying graph. Hence convex or concave elementary landscapes are flat for such triangle-free graphs (Theorem 11.16). That is another reason why the non-flat elementary landscapes induced by one-max, NAE3SATP and WPP on binary Hamming graphs in Section 11.3.2 cannot be convex nor concave.

Definition 11.7 ([**Sub-/super-**]**harmonic function** [16, 84])**.** Let a finite connected graph G with vertex set V and a function $f : V \to \mathbb{R}$. Then: (a) f is *harmonic*, if $\forall v \in V : (\mathbf{L}f)(v) = 0$; (b) f is *sub-harmonic*, if $\forall v \in V : (\mathbf{L}f)(v) \leq 0$; (c) f is *super-harmonic*, if $\forall v \in V : (\mathbf{L}f)(v) \geq 0$.

Proposition 11.3 (d**-metric convexity** $\implies$ **sub-harmonicity** [16])**.** Let a finite connected graph G with vertex set V and shortest-path metric d. If G has no triangles (i.e. subgraphs isomorphic to complete graphs K_3) nor vertices of degree less than two, then every d-metric convex function $f : V \to \mathbb{R}$ is sub-harmonic.

Remark 11.2. Under the same conditions in Proposition 11.3, d-metric concave functions are super-harmonic.

Lemma 11.3. For arbitrary finite connected graphs, only flat elementary landscapes are harmonic, sub-harmonic or super-harmonic.

Proof. Let $f : V \to \mathbb{R}$ be a graph Laplacian eigenfunction for an eigenvalue λ on any finite connected graph $G = (V, E)$, so that (G, f) is an elementary landscape by Definition 10.2: $(\mathbf{L}f)(v) = \lambda f(v)$, $\forall v \in V$. Then, either $\lambda = 0$ or $\lambda > 0$. If $\lambda = 0$, (G, f) is trivially a flat EL. If $\lambda > 0$, f is non-constant since f must be orthogonal to the order-zero graph Laplacian eigenfunction: $\langle f, \mathbf{1}\rangle = \sum_{v \in V} f(v) = 0$. Hence the entries of f must oscillate between positive and negative cancelling each other to satisfy $\langle f, \mathbf{1}\rangle = \sum_{v \in V} f(v) = 0$, for f cannot be the constant zero vector $(0, 0, \ldots, 0)$. That is, there exists at least one positive coordinate and one negative coordinate,

$\exists v_1, v_2 \in V : f(v_1) > 0$ and $f(v_2) < 0$. Therefore, f is not sub-harmonic nor super-harmonic since $\forall v \in V$: $(\mathbf{L}f)(v) = \lambda f(v) \not\leq 0$, and $(\mathbf{L}f)(v) = \lambda f(v) \not\geq 0$. Consequently, f is not harmonic neither. □

Remark 11.3 (To Lemma 11.3)**.** Biggs's Proposition 10.1 [7] proves that harmonic functions are constant on graphs, and Kiselman's Proposition 3.3 [84] shows that sub-harmonic functions are constant on graphs. Lemma 11.3 proves the same is true for elementary landscapes.

Theorem 11.16. Let G be any finite connected graph with vertex set V and shortest-path metric d but no triangles (i.e. subgraphs isomorphic to complete graphs K_3) nor vertices of degree less than two. If a function $f : V \to \mathbb{R}$ is d-metric convex and (G, f) is an elementary landscape, then f must be a constant graph Laplacian eigenfunction of G with eigenvalue $\lambda_p = 0$ of order $p = 0$.

Proof. From Proposition 11.3 it follows that every d-metric convex function f defined on such G is sub-harmonic. But Lemma 11.3 proves only flat elementary landscapes are sub-harmonic, regardless of G. Therefore, by Definition 10.2, f must be a constant graph Laplacian eigenfunction with eigenvalue $\lambda_p = 0$ of order $p = 0$. □

Note Theorem 11.16 does not apply, meaning that convex elementary landscapes need not be flat, if the graph has triangles or vertices of degree less than two as in the case of complete graphs. So Theorem 11.16 does not contradict Theorems 11.1–11.2 proving that all non-constant functions on all complete graphs are both elementary and convex.

11.4.2 Approximated Convexity

Let us consider perturbed fitness functions of the form $\tilde{f}(x) \stackrel{\text{def}}{=} f(x) + h(x)$ defined for all x in some domain X, where both f and h are real-valued functions defined on the same domain; f represents a given fitness function and h is bounded as $|h(x)| \leq \varepsilon$ for $\varepsilon \in [0, \infty)$. (In practice, any function encoded on a digital computer would be bounded.) The overall purpose of this section is to show and characterise various perturbed fitness functions defined on connected graphs, or graphic metric spaces, which are elementary and approximately (average or quasi) convex.

Recall Theorems 11.1–11.2 and Corollary 11.2 from Section 11.3.1 prove that every non-constant fitness function defined on any complete graph, or any finite d_0-metric space (X, d_0), constitutes a tight Fujiyama elementary landscape which is convex, average-convex and quasi-convex (and their concave counterparts). Clearly,

if we add arbitrary perturbations bounded by ε, the resulting landscape is also approximately (average and quasi) convex with tolerance ε by definition (see Section 9.3). This is true in particular for all pseudo-Boolean fitness functions defined earlier in Section 11.2, as Corollaries 11.9–11.10 state next.

Corollary 11.9 (To Theorem 11.1)**.** Let f be any of the one-max, needle-in-a-haystack, leading-ones, NAE3SATP, or WPP pseudo-Boolean fitness functions. Let also $\tilde{f}(x) \stackrel{\text{def}}{=} f(x) + h(x)$, where $h : \{0,1\}^n \to \mathbb{R}$ is any bounded function such that $|h(x)| \leq \varepsilon$ for all $x \in \{0,1\}^n$ and arbitrary $\varepsilon \in [0, \infty)$. Then, there exists an offset $c \in \mathbb{R}$ such that $(\{0,1\}^n, d_0, \tilde{f} - c\mathbf{1})$ is a tight Fujiyama elementary landscape.

Proof. It follows directly from Theorem 11.1. □

Corollary 11.10 (To Corollary 11.3)**.** Let f be any of the one-max, needle-in-a-haystack, leading-ones, NAE3SATP, or WPP pseudo-Boolean fitness functions. Let also $\tilde{f}(x) \stackrel{\text{def}}{=} f(x) + h(x)$, where $h : \{0,1\}^n \to \mathbb{R}$ is any bounded function such that $|h(x)| \leq \varepsilon$ for all $x \in \{0,1\}^n$ and arbitrary $\varepsilon \in [0, \infty)$. Then, $(\{0,1\}^n, d_0, \tilde{f})$ is approximately convex, approximately quasi-convex, and approximately average-convex with tolerance ε; and it is approximately concave, approximately quasi-concave, and approximately average-concave with tolerance $-\varepsilon$.

Proof. It follows directly from Definitions 9.4, 9.7 and 9.8, given that $(\{0,1\}^n, d_0, f)$ is convex, concave, quasi-convex, quasi-concave, average-convex and average-concave due to Corollary 11.3. □

In general, however, elementary landscapes do not remain elementary if their fitness function is arbitrarily perturbed. The next example for one-max fitness landscapes on three-dimensional binary Hamming graphs, which are elementary when zero-averaged, illustrates that even a slight perturbation breaks elementariness.

Example 11.1 (**Perturbed one-max landscape: not elementary**)**.** Consider the zero-averaged one-max fitness landscape $(\{0,1\}^3, d_{\text{H}}, f'_{\text{OneMax}})$, where $f'_{\text{OneMax}} \stackrel{\text{def}}{=} f_{\text{OneMax}} - \bar{f}_{\text{OneMax}}$ and $\bar{f}_{\text{OneMax}} = \frac{3}{2}$ is the global average, subject to a perturbation by a bounded function h as follows:

$$
\begin{array}{c}
\\
\texttt{000} \\
\texttt{001} \\
\texttt{010} \\
\texttt{011} \\
\texttt{100} \\
\texttt{101} \\
\texttt{110} \\
\texttt{111}
\end{array}
\begin{array}{c}
\tilde{f}_{\textsc{OneMax}} \\
\begin{bmatrix} -3/2 \\ -3/5 \\ -1/2 \\ 1/2 \\ -1/2 \\ 1/2 \\ 1/2 \\ 8/5 \end{bmatrix}
\end{array}
=
\begin{array}{c}
f'_{\textsc{OneMax}} \\
\begin{bmatrix} -3/2 \\ -1/2 \\ -1/2 \\ 1/2 \\ -1/2 \\ 1/2 \\ 1/2 \\ 3/2 \end{bmatrix}
\end{array}
+
\begin{array}{c}
h \\
\begin{bmatrix} 0 \\ -1/10 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1/10 \end{bmatrix}
\end{array}
\quad .
$$

Unlike the original $(\{0,1\}^3, d_{\mathrm{H}}, f'_{\textsc{OneMax}})$, the perturbed one-max fitness landscape $(\{0,1\}^3, d_{\mathrm{H}}, \tilde{f}_{\textsc{OneMax}})$ is not elementary with eigenvalue $\lambda = 2$ since

$$
\begin{array}{c}
\\
\begin{bmatrix}
3 & -1 & -1 & 0 & -1 & 0 & 0 & 0 \\
-1 & 3 & 0 & -1 & 0 & -1 & 0 & 0 \\
-1 & 0 & 3 & -1 & 0 & 0 & -1 & 0 \\
0 & -1 & -1 & 3 & 0 & 0 & 0 & -1 \\
-1 & 0 & 0 & 0 & 3 & -1 & -1 & 0 \\
0 & -1 & 0 & 0 & -1 & 3 & 0 & -1 \\
0 & 0 & -1 & 0 & -1 & 0 & 3 & -1 \\
0 & 0 & 0 & -1 & 0 & -1 & -1 & 3
\end{bmatrix}
\end{array}
\begin{array}{c}
\mathbf{L} \\
\begin{bmatrix} -3/2 \\ -3/5 \\ -1/2 \\ 1/2 \\ -1/2 \\ 1/2 \\ 1/2 \\ 8/5 \end{bmatrix}
\end{array}
=
\begin{array}{c}
\mathbf{L}\tilde{f}_{\textsc{OneMax}} \\
\begin{bmatrix} -29/10 \\ -13/10 \\ -1 \\ 1 \\ -1 \\ 1 \\ 9/10 \\ 33/10 \end{bmatrix}
\end{array}
\neq \lambda \cdot \tilde{f}_{\textsc{OneMax}} \quad ,
$$

where $\mathbf{L}$ is the graph Laplacian matrix of three-dimensional binary Hamming graphs.

Fortunately, it is possible to characterise a family of functions with which we can perturb elementary landscapes while preserving their elementariness. According to Lemma 11.4 below: any linear combination of a linear operator's eigenfunctions (like those of graph Laplacians) with the same eigenvalue will yield another eigenfunction for that operator and eigenvalue. Therefore, any elementary landscape of order p perturbed with any other associated graph Laplacian eigenfunction of the same order p, or linear combination thereof, will remain elementary. The following Example 11.2 illustrates it for one-max fitness landscapes on three-dimensional binary Hamming graphs, which would be also approximately average-convex and approximately average-concave according to Corollary 11.11 next.

Lemma 11.4. Let a linear operator $\mathbf{M} : \mathbb{R}^X \to \mathbb{R}^X$ on an inner product vector space $\mathbb{R}^X = \{f : X \to \mathbb{R}\}$ over any set X. Let $\varphi_1, \varphi_2 \in \mathbb{R}^X$ be any two eigenfunctions of $\mathbf{M}$ with identical eigenvalue λ. Then, $\varphi = a\varphi_1 + b\varphi_2$ is an eigenfunction of $\mathbf{M}$ with eigenvalue λ for any scalars $a, b \in \mathbb{R}$.

Proof. It follows from the additivity and scalar multiplication axioms of linear operators, namely $\mathbf{M}(f+g) = \mathbf{M}f + \mathbf{M}g$ and $c \cdot \mathbf{M}f = \mathbf{M}(cf)$ for all $f, g \in \mathbb{R}^X$:

$$\begin{aligned}\mathbf{M}\varphi &= \mathbf{M}(a\varphi_1 + b\varphi_2)\\ &= \mathbf{M}(a\varphi_1) + \mathbf{M}(b\varphi_2)\\ &= a\mathbf{M}\varphi_1 + b\mathbf{M}\varphi_2\\ &= a\lambda\varphi_1 + b\lambda\varphi_2\\ &= \lambda(a\varphi_1 + b\varphi_2) = \lambda\varphi \ .\end{aligned}$$

□

Remark 11.4. Lemma 11.4 proves for function vector spaces the property known as *closure under linear combinations* [113]. It was used by Langdon [87] several times during the analysis of various elementary landscapes though no proof is given.

Example 11.2 (**Perturbed one-max landscape: elementary**)**.** Consider the zero-averaged one-max fitness landscape $(\{0,1\}^3, d_{\mathrm{H}}, f'_{\textsc{OneMax}})$, where $f'_{\textsc{OneMax}} \stackrel{\text{def}}{=} f_{\textsc{OneMax}} - \bar{f}_{\textsc{OneMax}}$ and $\bar{f}_{\textsc{OneMax}} = \frac{3}{2}$ is the global average, subject to a perturbation by a bounded function h as follows:

$$\begin{array}{c} \\ \mathtt{000}\\ \mathtt{001}\\ \mathtt{010}\\ \mathtt{011}\\ \mathtt{100}\\ \mathtt{101}\\ \mathtt{110}\\ \mathtt{111}\end{array}\!\!\begin{array}{c}\tilde{f}_{\textsc{OneMax}}\\ \left[\begin{array}{c}-9/2\\ -1/2\\ -1\\ 3\\ -3\\ 1\\ 1/2\\ 9/2\end{array}\right]\end{array} = \begin{array}{c} f'_{\textsc{OneMax}}\\ \left[\begin{array}{c}-3/2\\ -1/2\\ -1/2\\ 1/2\\ -1/2\\ 1/2\\ 1/2\\ 3/2\end{array}\right]\end{array} + \begin{array}{c} h\\ \left[\begin{array}{c}-3\\ 0\\ -1/2\\ 5/2\\ -5/2\\ 1/2\\ 0\\ 3\end{array}\right]\end{array}$$

such that $h = 3\varphi + \frac{1}{2}\varphi'$ is a linear combination of two graph Laplacian eigenfunctions[1] $\varphi = (-1,0,0,1,-1,0,0,1)^{\mathrm{T}}$ and $\varphi' = (0,0,-1,-1,1,1,0,0)^{\mathrm{T}}$, as column vectors, with eigenvalue $\lambda = 2$ for a three-dimensional binary Hamming graph. Then,

[1] The eigenfunctions φ and φ' can be easily calculated with a modern computer algebra system like Mathematica.

$(\{0,1\}^3, d_{\mathrm{H}}, \tilde{f}_{\textsc{OneMax}})$ is elementary with eigenvalue $\lambda = 2$ since

$$\begin{array}{cccccccc} & & & \mathbf{L} & & & & \end{array} \quad \tilde{f}_{\textsc{OneMax}} \quad \mathbf{L}\tilde{f}_{\textsc{OneMax}}$$

$$\begin{bmatrix} 3 & -1 & -1 & 0 & -1 & 0 & 0 & 0 \\ -1 & 3 & 0 & -1 & 0 & -1 & 0 & 0 \\ -1 & 0 & 3 & -1 & 0 & 0 & -1 & 0 \\ 0 & -1 & -1 & 3 & 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & 3 & -1 & -1 & 0 \\ 0 & -1 & 0 & 0 & -1 & 3 & 0 & -1 \\ 0 & 0 & -1 & 0 & -1 & 0 & 3 & -1 \\ 0 & 0 & 0 & -1 & 0 & -1 & -1 & 3 \end{bmatrix} \begin{bmatrix} -9/2 \\ -1/2 \\ -1 \\ 3 \\ -3 \\ 1 \\ 1/2 \\ 9/2 \end{bmatrix} = \begin{bmatrix} -9 \\ -1 \\ -2 \\ 6 \\ -6 \\ 2 \\ 1 \\ 9 \end{bmatrix} = \lambda \cdot \tilde{f}_{\textsc{OneMax}} \quad ,$$

where $\mathbf{L}$ is the graph Laplacian matrix of three-dimensional binary Hamming graphs.

Corollary 11.11 (To Theorem 11.3)**.** Let $(\{0,1\}^n, d_{\mathrm{H}}, f'_{\textsc{OneMax}})$ be a zero-averaged one-max fitness landscape, where $f'_{\textsc{OneMax}} \stackrel{\text{def}}{=} f_{\textsc{OneMax}} - \bar{f}_{\textsc{OneMax}}$ and $\bar{f}_{\textsc{OneMax}} = \frac{n}{2}$ is global average. Let also $\tilde{f}_{\textsc{OneMax}}(x) \stackrel{\text{def}}{=} f'_{\textsc{OneMax}}(x) + h(x)$, where and $h : \{0,1\}^n \to \mathbb{R}$ is a bounded function such that $|h(x)| \leq \varepsilon$ for all $x \in \{0,1\}^n$ and some $\varepsilon \in [0, \infty)$. Then:

(a) $(\{0,1\}^n, d_{\mathrm{H}}, \tilde{f}_{\textsc{OneMax}})$ is approximately average-convex with tolerance ε and approximately average-concave with tolerance $-\varepsilon$.

(b) If h is a graph Laplacian eigenfunction with eigenvalue $\lambda = 2$ for the binary Hamming graph $H(n, 2)$, then $(\{0,1\}^n, d_{\mathrm{H}}, \tilde{f}_{\textsc{OneMax}})$ is a tight Fujiyama elementary landscape also.

Proof. (a) Theorem 11.3 proves $(\{0,1\}^n, d_{\mathrm{H}}, f_{\textsc{OneMax}})$ is average-convex and average-concave. Therefore, by Definition 9.8, $\tilde{f}_{\textsc{OneMax}}$ is a $(d_{\mathrm{H}}, \varepsilon)$-average-convex function and a $(d_{\mathrm{H}}, -\varepsilon)$-average-concave function. (The offset $\bar{f}_{\textsc{OneMax}}$ makes effectively no difference because all fitness values of $\tilde{f}_{\textsc{OneMax}}$ would shift by the same amount.)

(b) It follows directly from Lemma 11.4 and that zero-averaged one-max fitness landscapes $(\{0,1\}^n, d_{\mathrm{H}}, f'_{\textsc{OneMax}})$ are tight Fujiyama elementary landscapes with eigenvalue $\lambda = 2$ (see Section 11.3.2). □

Lemma 11.4 showed that perturbed fitness landscapes of the form $f + h$, where f is elementary and h is any linear combination of elementary landscapes for the

same eigenvalue as f, characterises a family of elementary landscapes which can be approximately (average or quasi) convex depending on whether f is convex, average-convex or quasi-convex. The following Theorem 11.17 characterises such perturbed elementary landscapes in terms of their graph Laplacian eigenvalue, associated neighbourhood function and average fitness value of neighbours.

Theorem 11.17. Let any function $f : V \to \mathbb{R}$ defined on the vertex set V of an arbitrary finite connected graph G. Let also $h : V \to \mathbb{R}$ be any bounded function, with the same domain V as f, such that $|h(v)| \leq \varepsilon$ for all $v \in V$ and arbitrary $\varepsilon \in [0, \infty)$. Then, the function $\tilde{f}(v) \stackrel{\text{def}}{=} f(v)+h(v)$ is a graph Laplacian eigenfunction of G with eigenvalue λ if and only if f and h satisfy:

$$1 - \frac{\lambda}{|N(v)|} = \frac{\underset{u \in N(v)}{\text{avg}} \{f(u)\} + \underset{u \in N(v)}{\text{avg}} \{h(u)\}}{f(v) + h(v)} \ , \ \forall v \in V, \tag{11.6}$$

where $|N(v)|$ is the number of neighbours adjacent to v in G, and $\underset{u \in N(v)}{\text{avg}} \{f(u)\}$ and $\underset{u \in N(v)}{\text{avg}} \{h(u)\}$ are the arithmetic mean values of f and h over all neighbours u of v.

Proof. Using Equation 10.3, $\tilde{f}$ is a graph Laplacian eigenfunction with eigenvalue λ if and only if for all $v \in V$: $(\mathbf{L}\tilde{f})(v) = |N(v)|\, \tilde{f}(v) - \sum_{u \in N(v)} \tilde{f}(u) = \lambda \tilde{f}(v)$. This is true if and only if the following series of equalities hold:

$$|N(v)| \Big(f(v) + h(v)\Big) - \sum_{u \in N(v)} \Big(f(u) + h(u)\Big) = \lambda \Big(f(v) + h(v)\Big),$$

$$|N(v)| \Big(f(v) + h(v)\Big) - \lambda \Big(f(v) + h(v)\Big) - \left(\sum_{u \in N(v)} f(u)\right) - \left(\sum_{u \in N(v)} h(u)\right) = 0,$$

$$f(v) + h(v) - \frac{\lambda}{|N(v)|} \Big(f(v) + h(v)\Big) - \frac{1}{|N(v)|} \left[\left(\sum_{u \in N(v)} f(u)\right) - \left(\sum_{u \in N(v)} h(u)\right)\right] = 0,$$

$$\Big(f(v) + h(v)\Big) \left(1 - \frac{\lambda}{|N(v)|}\right) - \underset{u \in N(v)}{\text{avg}} \{f(u)\} - \underset{u \in N(v)}{\text{avg}} \{h(u)\} = 0,$$

from which Equation 11.6 follows. □

Remark 11.5 (To Theorem 11.17)**.** For D-regular graphs the number of neighbours is constant: $|N(v)| = D$, $\forall v \in V$. So Equation 11.6 becomes:

$$1 - \frac{\lambda}{D} = \frac{\underset{u \in N(v)}{\text{avg}} \{f(u)\} + \underset{u \in N(v)}{\text{avg}} \{h(u)\}}{f(v) + h(v)}, \ \forall v \in V \ . \tag{11.7}$$

Remarkably, Stadler [137] proves that $1 - \frac{\lambda}{D}$ is the nearest-neighbour fitness correlation of any elementary landscape with eigenvalue λ on any D-regular graph.

Remark 11.6 (To Theorem 11.17)**.** Chicano and others [20] prove a closely related identity: $\underset{u \in N(v)}{\text{avg}} \{f(u)\} = f(v) - \frac{\lambda}{D}(c - f(v))$, for some constant $c \in \mathbb{R}$. But it has not been used to characterise perturbed elementary landscapes as Equation 11.6.

Although Theorem 11.17 is an immediate consequence of the Definition 10.2 of elementary landscapes, from Equation 11.6 it is clearer to see that not all approximately (average or quasi) convex landscapes of the form $f + h$ will be elementary if we allow an arbitrary perturbation h because there is no guarantee $f + h$ will respect Equation 11.6 even if f itself is elementary. By contrast, we are guaranteed Equation 11.6 or Equation 11.7 is fulfilled if elementary landscapes are perturbed by linear combinations of elementary landscapes conforming to Lemma 11.4, as illustrated in the next example.

Example 11.3. Recall from the previous Example 11.2 the zero-averaged one-max fitness landscape $(\{0,1\}^3, d_{\text{H}}, f'_{\text{OneMax}})$ perturbed by a bounded function h, where

$$f'_{\text{OneMax}} = \Big(\overset{000}{-\frac{3}{2}}, -\frac{1}{2}, -\frac{1}{2}, \frac{1}{2}, -\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \overset{111}{\frac{3}{2}}\Big)$$

and

$$h = \Big(-3, 0, -\frac{1}{2}, \frac{5}{2}, -\frac{5}{2}, -\frac{1}{2}, 0, 3\Big) \quad ,$$

both of which are elementary with eigenvalue $\lambda = 2$. Since $(\{0,1\}^3, d_{\text{H}})$ corresponds to a three-dimensional binary Hamming graph $H(3,2)$, and it is a 3-regular graph, the left-hand side of Equation 11.7 is

$$1 - \frac{\lambda}{D} = 1 - \frac{2}{3} = \frac{1}{3} \quad .$$

For the right-hand side of Equation 11.7, recall $H(3,2)$ is described by the single-bit flip neighbourhood function $\textsc{Bitflip}(1)$. Then, choosing vertex $\mathtt{100}$, for instance, we have

$$\frac{\underset{u \in \textsc{Bitflip}(1)(\mathtt{100})}{\text{avg}} \{f'_{\text{OneMax}}(u)\} + \underset{u \in \textsc{Bitflip}(1)(\mathtt{100})}{\text{avg}} \{h(u)\}}{f'_{\text{OneMax}}(\mathtt{100}) + h(\mathtt{100})} = \frac{\left(-\frac{1}{6}\right) + \left(-\frac{5}{6}\right)}{\left(-\frac{1}{2}\right) + \left(-\frac{5}{2}\right)} = \frac{1}{3} \quad .$$

Hence Equation 11.7 holds for the vertex `100`. It is not difficult to see the same is true of other vertices.

11.5 Discussion

A major aim of unifying the GF and ELT is to know whether elementary landscapes associated with relevant combinatorial optimisation problems correspond to any of the abstract convex landscape classes where geometric-crossover EAs expectedly perform well. Pursuing this aim fundamentally entails a classification of abstract convex elementary landscapes. To investigate its viability and potential implications, this chapter developed the first theoretical foundations of abstract convex elementary landscapes by addressing two fundamental questions:

- Are the classes of elementary landscapes and abstract convex fitness landscapes disjoint? If not, which specific classes do overlap? (Question 11.1)
- How does the abstract convexity of an elementary landscape relate to its eigenfunction order or position in the Laplacian spectrum? (Question 11.2)

The discussion of Questions 11.1 and 11.2 alongside the results obtained in previous Sections 11.3 and 11.4 is split into: viability of fitness landscape classification (Section 11.5.1), spectral identification of abstract convexity (Section 11.5.2), and problem difficulty and EAs (Section 11.5.3).

11.5.1 Viability of Fitness Landscape Classification

Question 11.1 concerns three mutually-exclusive cases that may occur in a classification based on the abstract convex landscapes and elementary landscapes classes: (a) they are disjoint classes, thus no landscape is abstractly convex and elementary; (b) they are the same class, thus all abstract convex landscapes are elementary and vice versa; and (c) they are neither disjoint nor the same class, thus some but not all landscapes are abstractly convex and elementary. Clearly, only case (c) would be viable to develop a classification by axiomatically comparing both classes, for they would share none or all properties in the extreme cases (a) and (b).

On finite d_0-metric spaces, or complete graphs, case (b) is effectively true. Sections 11.3.1 and 11.4.2 show every non-constant fitness function induces a tight Fujiyama elementary landscape that is also: (approximately) convex, (approximately) quasi-convex, (approximately) average-convex as well as their concave coun-

terparts. Obviously, flat landscapes are elementary and belong to those abstract convex classes.

On binary Hamming metric spaces, or binary Hamming graphs, it is clear from the summary in Table 11.3 that the extreme cases (a) and (b) are false, meaning case (c) is true. The one-max fitness function induces an average-convex, average-concave, and tight Fujiyama elementary landscape; in addition, certain perturbed one-max fitness functions induce approximately average-convex, approximately average-concave and tight Fujiyama elementary landscapes (Corollary 11.11). So case (a) is false. By contrast, the needle and leading-ones fitness functions induce quasi-concave landscapes but these are not elementary; conversely, NAE3SATP and WPP induce elementary landscapes but are not convex, quasi-convex, average-convex nor their concave counterparts. So case (b) is false too.

However, binary Hamming graphs are not totally exempt from other extreme situations: for the class of triangle-free graphs with vertex degree two or higher, of which all non-trivial binary Hamming graphs are members, there is no convex nor concave elementary landscape except a flat landscape (Theorem 11.16).

Therefore, abstract convex elementary landscapes do exist despite abstract convexity and elementariness bear no relationship at first sight. But extreme cases may occur depending on the metric space or graph; for instance, they may comprise all fitness functions or only constant fitness functions. To find other examples of abstract convex elementary landscapes, hopefully more interesting than (perturbed) one-max, while avoiding degenerate cases, future work should then narrow down the classes of abstract convex landscapes and elementary landscapes proposed by Moraglio [101] and Stadler [136] respectively. (Relaxing their definitions would not prevent degenerate cases.) The main difficulty in doing so is that ELT was not conceived as a general framework to systematically classify the various types of abstract convexity possibly occurring in elementary landscapes. The next Section 11.5.2 suggests how it may be achieved.

11.5.2 Spectral Identification of Abstract Convexity

Stadler and others [5, 136, 138] classified elementary landscapes by their smoothness (understood as correlation between fitness values) and showed that different eigenvalues determine different degrees of smoothness. Moreover, on binary Hamming graphs, smoothness decreases as the elementary landscape's position in the Laplacian spectrum moves towards higher-order eigenvalues, and Stadler conjectured the same could be true for other graphs [137, 139]. Question 11.2 goes beyond, asking

if some general rule relates the abstract convexity of Laplacian eigenfunctions, that is elementary landscapes, to a specific position in the Laplacian spectrum. This question is a stepping stone to Conjecture 11.1, discussed later in Section 11.5.3 in connection with problem difficulty and EAs.

Conjecture 11.1. Globally convex elementary landscapes correspond to low-order graph Laplacian eigenfunctions: the higher the order, the less globally convex.

Part of Conjecture 11.1 is shown by Theorem 11.15 indicating one general rule: for an arbitrary finite connected graph, every non-flat elementary landscape either convex, concave, quasi-convex, or quasi-concave, corresponds in general to an order-one graph Laplacian eigenfunction associated with the smallest non-zero eigenvalue in the spectrum. Of course, flat elementary landscapes are always order-zero graph Laplacian eigenfunctions, (quasi-)convex and (quasi-)concave. This rule has the following significant implications.

First, it shows that if we know a fitness landscape is elementary, or find it analytically with the methodology developed by Chicano, Whitley and Alba [23], then we know straightaway whether it can be (quasi-)convex or (quasi-)concave solely by looking at its spectrum. If the eigenvalue's order is greater than one, then such fitness landscape is neither (quasi-)convex nor (quasi-)concave. It agrees also with what Stadler conjectured earlier precisely because non-flat elementary landscapes corresponding to order-one eigenvalues would be the smoothest. Even detecting (quasi-)convexity algorithmically might be possible: although it is an NP-hard problem for general functions [1], in some cases the convexity of discrete functions can be detected efficiently [111].

Secondly, the vast majority of elementary landscapes reported [85], including those of NAE3SATP and WPP as well as TSP and spin-glass for typical neighbourhoods, are graph Laplacian eigenfunctions of order greater or equal than two. So the vast majority of known elementary landscapes are neither (quasi-)convex nor (quasi-)concave. Hence, if we want to use elementary landscapes to investigate the question, suggested by Moraglio and Sudholt [104], of which non-toy problems have associated quasi-concave landscapes, Theorem 11.15 forces us to restrict to weakly unimodal[2] Fujiyama elementary landscapes or a subclass thereof. That is, order-one graph Laplacian eigenfunctions satisfying Definition 10.8. One potential example is the linear assignment problem since its fitness function induces a

[2]Weak unimodality is always guaranteed for Fujiyama elementary landscapes defined on graphs resulting from Cartesian products of paths, cycles and complete graphs due to Tlusty's Corollary 2.6 [145] (see Proposition 10.7). Unlike Theorem 11.16, Proposition 10.7 does not necessarily lead to flat landscapes because quasi-convex landscapes are a superclass of convex landscapes.

Fujiyama elementary landscape on the Cayley graph given by transpositions (i.e. swap of two positions in a permutation) [85, 121] though no proof of quasi-convexity nor quasi-concavity has been shown yet. This example has not been explored in this chapter because elementary landscapes on Cayley graphs involve more specialised mathematics from group theory [136].

It is tentative to speculate that the previous rule for identifying (quasi-)convex elementary landscapes extends to average-convex or approximately average-convex elementary landscapes, given that it applies in the case of one-max and certain perturbed one-max landscapes on binary Hamming graphs (see Corollary 11.11). To confirm if the rule is valid in other cases, further research needs to prove if there actually exist (approximately) average-convex elementary landscapes for eigenvalues of order greater or equal than two, taking into account:

- Unlike (quasi-)convex functions, (approximately) average-convex functions may not be weakly unimodal nor have geodesically convex sub-level sets (see Proposition 9.3), which was key for Theorem 11.15 supporting the aforesaid rule.
- Approximately average-convex landscapes of the form $\tilde{f} = f + h$, where f is an average-convex function and h is an arbitrary bounded perturbation, are not elementary in general because (even if f is elementary) h can easily violate Theorem 11.17 which all elementary landscapes (perturbed or not) fulfil.

11.5.3 Problem Difficulty and Evolutionary Algorithms

In spite of the outstanding efforts made by Stadler, Wagner and many others [51, 131, 139, 140, 150] to show that (recombination P-structure) elementary landscapes have distinctive features beneficial for EAs performance, the truth is no formal performance or runtime analysis of any specific population-based EA class has been accomplished yet for the class of (recombination P-structure) elementary landscapes. It should be said though Klemm and Stadler [85] emphasised that the geometrical, spectral and statistical properties of such landscapes would play a crucial role in a theory that predicts the performance of EAs optimising them.

The significance of this chapter's results (Sections 11.3–11.4) and discussion (Sections 11.5.1–11.5.2), which culminate in Conjecture 11.1, is they partially address what Klemm and Stadler mentioned by proving the existence of globally convex elementary landscapes as well as key relationships with the Laplacian spectrum to characterise them, thanks to the abstract convex landscape classes proposed by Moraglio [100, 101]. An underlying implication is that the order of globally elementary landscapes, namely the position of globally convex elementary landscapes

in the spectrum ignoring eigenvalue multiplicities, might be an indicator of problem difficulty for certain geometric-crossover EAs or recombination P-structure EAs based on the following observations.

Globally convex elementary landscapes can be defined on finite graphic metric spaces with a shortest-path metric distance, conforming to the prerequisites stated in Section 11.2.2. Every such metric space is associated with a specific class of geometric-crossover EAs searching within it (see Section 9.1.1) and, in particular, is associated with geometric recombination P-structure EAs since these are a subclass of the former (see Chapter 8). Moreover, geometric recombination P-structures, as geometric crossovers, are highly local crossovers (see Chapter 5), which Rothlauf [123, 124], Droste and Wiesmann [38] deem decisive in favouring exploitative search to outperform pure random search. Non-geometric recombination P-structures instead are generally not associated with metrics because they are not geometric crossovers.

Problem difficulty comes into play taking into account Goldberg [54, 55] and Rothlauf [123, 124] show that pseudo-Boolean fitness functions which decompose into linear combinations of low-order Walsh basis functions are usually 'easier' to optimise by recombination-based EAs (without mutation) compared with higher order ones. This applies to elementary landscapes, globally convex or not, defined on binary Hamming graphs because they are Walsh functions too (see Section 10.2.2), and it agrees with Stadler and Wagner [139] in that lower-order elementary landscape are smoother. But Goldberg and Rothlauf warn us that the order of Walsh functions is not a general indicator of problem difficulty: it may overestimate or underestimate the difficulty of some problems. However, these misestimates seem unclear for elementary landscapes: none of the counterexamples considered by Goldberg [54, 55] and Rothlauf [123, 124] were elementary landscapes but rather linear combinations of elementary landscapes of different orders, specifically linear combinations of Walsh functions.

For instance, Moraglio and Sudholt [104] prove that a geometric-crossover EA optimises the leading-ones fitness function on n-dimensional binary Hamming graphs in $\mathcal{O}(n \log n)$ fitness evaluations. One may argue that the Walsh order of leading-ones overestimates the difficulty since it is a sum of Walsh functions up to order n (Theorem 11.9), but that is exactly why leading-ones on binary Hamming graphs is not elementary to begin with! An extremely degenerate counterexample as to why the order of globally convex elementary landscapes may not be a problem difficulty indicator is that recombination-based EAs using the identity crossover, where offspring always equal parents, would perform 'poorly' at optimising virtually

any non-constant fitness function. Yet any non-constant fitness function on the complete graph induced by identity crossover defines a globally convex Fujiyama elementary landscape of order one (see Section 11.3.1), which is considered 'easy', thereby underestimating the difficulty. Then again, identity crossover and complete graphs are degenerate cases.

Finally, the position of (globally convex) elementary landscapes in the Laplacian spectrum also gives upper-bounds of the total number of discrete nodal domains [8, 30], namely 'clusters' of candidate solutions (including local optima) either above or below average fitness (see Section 10.4.1). This feature was deemed of central importance by Stadler [136] and is not that far from the local optima networks whereby Ochoa and Veerapen [109] provide a more accurate empirical view of the big valley hypothesis. Clearly, the upper-bounds in Propositions 10.4 and 10.5 increase with the position or order of elementary landscapes, meaning that the number of discrete nodal domains would increase when elementary landscapes become less globally convex in principle.

11.6 Conclusion

Past empirical studies observe certain local search algorithms and GAs yield higher quality approximate solutions for various important combinatorial problems when their associated fitness landscapes have a globally convex or big valley structure. These concern ELT and the GF. On the one hand, ELT showed such landscapes can be found in the class of elementary landscapes, even though ELT missed the link with global convexity and population-based EAs. On the other hand, the GF defined abstract convex landscape classes, with various notions of global convexity, where geometric-crossover EAs expectedly perform well; however, GF left unclear which specific problems and associated landscapes belong to those convex landscape classes.

My original contribution is to integrate the above two separate lines of research coming from the GF and ELT by characterising a general class of combinatorial fitness landscapes shared between the GF and ELT. That is, abstract convex elementary landscapes. These are always guaranteed to be elementary, unlike the classes of abstract convex landscapes in the GF (Chapter 9), and guaranteed to be abstractly convex at the same time, unlike elementary landscapes in ELT (Chapter 10). Abstract convex elementary landscapes offer an opportunity to know whether geometric-crossover EAs, including geometric recombination P-structure EAs (Chapter 8), would perform well on certain elementary landscapes by

proving they are globally convex. This matters because it means that abstract convex elementary landscapes can help us better understand on which specific combinatorial problems do certain geometric-crossover EAs perform well and, at the same time, gain insight into the spectral properties of elementary landscapes in relation with population-based EAs and abstract convexity. Both of which are current open challenges in the GF and ELT respectively.

However, carrying the above research programme on abstract convex elementary landscapes is easier said than done, for abstract convex landscapes in the GF and elementary landscapes in ELT are significantly different classes of fitness landscapes without any apparent similarities. This chapter establishes theoretical foundations of abstract convex elementary landscapes through a fitness landscape classification of well known pseudo-Boolean combinatorial problems against the classes of abstract convex landscapes and elementary landscapes. Then it characterises the abstract convexity of elementary landscapes more generally in terms of Laplacian eigenvalues; in contrast with ELT, which focused on smoothness rather than convexity, and with the GF, which never analysed elementariness as a feature of abstract convex landscapes.

Although non-degenerate abstract convex elementary landscapes arise only under limited circumstances, this chapter proved their existence and gave insight into their classification to find more interesting examples. Furthermore, it justified that abstract convex elementary landscapes mainly correspond to low-order graph Laplacian eigenfunctions, which can also explain why some geometric-crossover EAs would perform well on them. Put differently, this suggests abstract convex elementary landscapes are a viable means to develop (partly based on ELT) fitness landscape analysis within the GF which was lacking. Whether abstract convex elementary landscapes are actually ‘practical’ remains questionable and calls for a more exhaustive study about them; this chapter serves as a safe starting point.

Chapter 12

Conclusions

A great challenge to develop a general theory of EC, independent of specific problems and representation of solutions, is posed by the question: why does a given class of EAs perform better (or worse) than others on a given class of optimisation problems? To rigorously answer this question is not simple because it concerns various challenging aspects of EC such as analysing the population behaviour of EAs, analysing their (runtime) performance, and analysing fitness landscapes that relate EAs and optimisation problems. Despite being motivational and not answered in this thesis, the aforesaid question concerns two general theories in EC which have addressed it differently and which are central to this thesis. They are the geometric framework (GF) of EAs and elementary landscapes theory (ELT): the former focused on the design, population behaviour and performance of EAs, whereas the latter focused more on the analysis of combinatorial landscapes. This thesis explored if and how the GF and ELT can be unified; that is, how their strengths to analyse EAs and landscapes complement each other and can be integrated within one cohesive framework, a general theory characterising which axiomatic classes of crossovers, evolutionary search and landscapes are shared between the GF and ELT. As no such general theory was explored before, it remained unclear if and how it is theoretically possible to unify the GF and ELT. Particularly, this thesis addressed the following general research questions:

1. How can geometric crossovers, proposed in the GF, and recombination P-structures, proposed in ELT, be unified?

2. How can geometric-crossover EAs, proposed in the GF, and recombination P-structure random walks, proposed in the ELT, be unified?

3. How can abstract convex fitness landscapes, proposed in the GF, and

elementary landscapes, proposed in the ELT, be unified?

4. What are the consequences of each of the previous unifications and how do they contribute to unify the GF and ELT under a common mathematical framework?

The original contribution of this thesis is to formally show a three-fold unification between the GF and ELT, contributing towards question (4), and to provide a comprehensive revision of the foundations of GF and ELT which support the unification. That is,

- a *classification of crossovers* (Chapter 5) focusing on question (1);
- a *qualitative framework for abstract convex search of evolutionary algorithms* (Chapter 8) focusing on question (2); and,
- *abstract convex elementary landscapes* (Chapter 11) focusing on question (3).

12.1 Contributions

This section discusses each of the three unifications above. Section 12.2 discusses limitations of this thesis and concludes it with suggestions for future research.

12.1.1 A Classification of Crossovers

To unify the GF and ELT, this thesis began by introducing two general crossover classes independent of problems and solution representations: the class $\mathcal{GX}$ that includes all geometric crossovers (Chapter 3) and the class $\mathcal{RP}$ that includes all recombination P-structures (Chapter 4). My first major contribution is a formal crossover classification based on the mathematical axioms that define $\mathcal{GX}$ and $\mathcal{RP}$ (Chapter 5). This crossover classification showed $\mathcal{GX}$ and $\mathcal{RP}$ share the subclass $\mathcal{RP}$-geometric, which includes all crossovers that are both geometric and recombination P-structures. This is significant because, unlike the GF as proposed by Moraglio [100] and ELT as proposed by Stadler [136] separately, I make provably clear in Chapter 5 that:

- The GF and ELT share a common class of crossovers, namely $\mathcal{RP}$-geometric, which are potentially useful in practice and can be designed in a principled manner across different representations and problems associated with shortest-path distances based on finite connected graphs. This suggests principled design of recombination P-structures is an essential part of ELT, even though

previous research in ELT never addressed it [18, 51, 131, 139, 150]. Principled design of geometric crossovers was already formalised by Moraglio [100], but it remained unclear whether principled design would be possible also for recombination P-structures (as for $\mathcal{RP}$-geometric) because the GF never analysed recombination P-structures.

- There exist recombination P-structures, like geometric recombination P-structures, which can exploit problem-specific features related with distances such as high locality [38, 124] and abstract convexity [104, 146] to achieve better performance than random search by avoiding NFL [93, 155].

Although Moraglio [100] distinguished between geometric crossovers and non-geometric crossovers, an axiomatic and comprehensive crossover classification was missing in the GF. In fact, his focus is more the formalisation of geometric crossovers than classifying which crossovers are geometric or not. Chapter 5 not only provides a crossover classification for the first time but also key insight into various properties related to geometric crossovers, namely high locality and the inbreeding properties, beyond the work of Moraglio [100, 103]. That is, without the help of the crossover classification it would be less clear why $\mathcal{RP}$-geometric should be the crossover subclass shared by $\mathcal{GX}$ and $\mathcal{RP}$ since the frontier between geometric crossovers and recombination P-structures is not obvious to determine:

- Any geometric crossover, including those in $\mathcal{RP}$-geometric, is highly local even though not all highly local crossovers are geometric as shown in Chapter 5. For example, the intersecting-Hamming-balls crossover is a recombination P-structure that is highly local but non-geometric. In other words, geometricity and high locality are not equivalent properties of crossovers (see Section 5.5.1).

- Any geometric crossover, including those in $\mathcal{RP}$-geometric, is associated with a metric space and fulfils the inbreeding properties [103]. But there exist a subclass of recombination P-structures which is not defined on metric spaces necessarily and still fulfils all the inbreeding properties of geometric crossovers. That is, strict size-monotonic recombination P-structures (see Section 5.5.2), which have been ignored by previous research in ELT [18, 51, 139, 150] and are clearly outside Moraglio's scope of geometric crossovers [100].

Above all, the crossover classification is a formal system that helps us to rigorously organise and compare the axiomatic properties of different crossovers, and classes thereof, with respect to geometric crossovers and recombination P-structures

across different problems and solution representations. The alternative to formal classifications, as the one presented here, are classifications possibly based on historical, subjective or superficial aspects of EAs [9, 122][36, ch. 9][52, ch. 17]. These are, therefore, informal classifications which can be helpful but not as a theoretical basis to unify the GF and ELT.

12.1.2 A Qualitative Framework for Abstract Interval Convex Search of Evolutionary Algorithms

In Chapter 8, I develop a qualitative framework for abstract interval convex evolutionary search by extending the crossover classification I laid in Chapter 5. The extension introduces a class of crossovers $\mathcal{I}_{fin}$, which are described by finite interval operators [18, 146] and need not coincide with Moraglio's geometric crossovers (Section 3.2) nor Stadler's recombination P-structures (Section 4.2.1): $\mathcal{I}_{fin}$ is a superclass of all (geometric or non-geometric) recombination P-structures, and it includes other forms of crossover similar to the macro-mutations present in headless-chicken crossover random walks [73]. This leads to a generalised class of EAs, corresponding to $\mathcal{I}_{fin}$, called formal interval EAs: an evolutionary search model common to the GF and ELT, whereby I generalise geometric-crossover EAs (Chapter 6) and recombination P-structure random walks (Chapter 7). As a result, Chapter 8 presents a framework that broadens but differs from related work of Moraglio [100, 101] and Stadler [140, 150] in various significant ways:

- First, abstract convex evolutionary search indeed can be generalised to a form not necessarily limited only to geometric-crossover EAs (without mutation) as the GF originally conceives (Section 6.3). That is, abstract *interval* convex evolutionary search holds for any formal interval EA even if mutation (e.g. Hamming ball-mutation segment) or non-geometric crossovers (e.g. intersecting-Hamming-balls) in the class $\mathcal{I}_{fin}$ are used. This may seem counter-intuitive since, according to Moraglio [101], abstract convex evolutionary search cannot occur if mutation or non-geometric crossovers are present. However, Moraglio stated it for the particular case of geodesic convexities, ignoring other abstract convexities altogether since geometric crossovers can always be associated with geodesic convexities (Section 6.2). In Chapter 8, I shift from Moraglio's said viewpoint by taking non-geodesic convexities into account as well and showing consequently that: each crossover in $\mathcal{I}_{fin}$ induces its own abstract interval convexity (Section 8.3) and need not be a geodesic convexity as for geometric crossovers.

- Unlike recombination P-structure random walks, which are a form of headless-chicken crossover random walk [73, 140, 150], formal interval EAs provide ELT with an actual population-based EA model for any choice of recombination P-structure: recombination P-structure EAs. As shown in Section 8.4, their abstract population behaviour is also described by abstract interval convex evolutionary search regardless of specific problems and solution representations. This suggests a theory of recombination P-structure EAs is a conceivable extension of ELT, in which case ELT is more than a theory of combinatorial landscapes as proposed by Stadler [136]. However, it would not be based on transition probability matrices associated with recombination P-structures [85, 140] (see Section 7.3) but abstract convexity as Section 8.3 explained in relation with the extended crossover classification.

- The GF and ELT share a class of EAs without mutation (i.e. geometric recombination P-structure EAs) where abstract *interval* convex evolutionary search and Moraglio's abstract convex evolutionary search are provably equivalent. This, however, is not evident from the work of Moraglio [101, 104] nor Stadler [139, 140, 150] because abstract convex evolutionary search never was analysed on any class of recombination P-structures (or generalisations thereof). Whether the aforesaid equivalence holds or not for formal interval EAs, other than geometric recombination P-structure EAs, depends on the underlying crossover: Section 8.4.1 illustrated some counterexamples and an exception for symmetric Davis's order crossover. Chapter 8 shows $\mathcal{I}_{fin}$ and $\mathcal{RP}$ are more general than geometric crossovers in the sense that they include non-geometric crossovers and can induce non-geodesic convexities (possibly degenerate), whereas geometric crossovers induce geodesic convexities. As shown in Section 8.4, non-geodesic convexities can lead to an abstract interval convex evolutionary search that is clearly *not* Moraglio's abstract (geodesically) convex evolutionary search.

Overall, Chapter 8 presents a framework to systematically classify the abstract interval convex evolutionary search induced by formal interval EAs depending on different crossovers in the crossover classification. This framework not only integrates the GF and ELT with each other but extends them as none of them originally developed formal means to classify the abstract behaviour of population-based EAs with crossover. Others [39, 63, 116] also proposed formal hierarchies, relating EAs to their performance on fitness landscapes, but not for geometric crossovers nor recombination P-structures. The next logical step seems, therefore, to relate EA

classes based on geometric crossovers or recombination P-structures to performance on fitness landscapes; that is, the following contribution in Section 12.1.3.

12.1.3 Abstract Convex Elementary Landscapes

Making meaningful statements about the performance of a class of EAs shared between the GF and ELT, such as geometric recombination P-structure EAs, requires a corresponding class of fitness landscapes and associated problems to be determined. NFL [68, 155] prevails otherwise. To do so, I characterise in Chapter 11 a class of combinatorial landscapes that can be abstractly convex (Chapter 9) and elementary (Chapter 10) simultaneously, thus shared between the GF and ELT, across problems and representations. That is, abstract convex elementary landscapes. These clearly differ from Moraglio's abstract convex landscapes [101], which need not be elementary in general, and from Stadler's elementary landscapes [136, 139], which need not be abstractly convex in general. Although the classes of abstract convex landscapes and elementary landscapes are seemingly disjoint, for they involve significantly different formalisations in the GF and ELT, in Chapter 11 I identified certain conditions where abstract convex elementary landscapes are conceivable and revealed that:

- There actually exist non-degenerate examples of abstract convex elementary landscapes. In particular, Chapter 11 showed that any one-max fitness function on binary Hamming graphs is an average-convex and average-concave elementary landscape. Indeed, the same holds for any one-max function that is perturbed by (bounded) arbitrary linear combinations of elementary landscapes with the same Laplacian eigenvalue.

This is significant because no previous theoretical research in ELT [5, 23, 35, 136, 139, 140, 150] analysed abstract convexity as a feature of elementary landscapes nor noticed abstract convex elementary landscapes provably exist. Yet empirical studies [10, 46] did observe a globally convex (or 'big valley') structure [10, 83, 109] in certain combinatorial problems associated with elementary landscapes (e.g. TSP or graph bipartition) [56, 85], which also benefited the performance of certain local search algorithms and GAs [10, 46]. In fact, these experiments partly motivated Moraglio [100, 101, 104] to propose the classes of abstract convex landscapes. However, Moraglio never analysed elementariness as a feature of abstract convex landscapes; Chapter 11 does for the first time. Why did abstract convex elementary landscapes pass unnoticed until now? The main reason is that abstract convex

elementary landscapes are a more restricted class than elementary landscapes and abstract convex landscapes. This is justified in Chapter 11 by proving:

- All convex, concave, quasi-convex or quasi-concave elementary landscapes correspond to graph Laplacian eigenfunctions of order less than two for any finite connected graph. It is true also for the average-convex (and average-concave) elementary landscapes in the case of (perturbed) one-max functions mentioned earlier, which correspond a graph Laplacian eigenvalue of order one.

The result above is relevant for two reasons. On the one hand, it reveals that the typical elementary landscapes reported in [85] associated with combinatorial problems like TSP, NAE3SATP or max-cut, whose Laplacian eigenvalue is of order two or higher, are *not* (quasi-)convex nor (quasi-)concave. On the other hand, it reveals the *exact* location in the Laplacian spectrum where potential examples can be found since: non-constant (quasi-)convex or (quasi-)concave elementary landscapes correspond to the smallest non-zero eigenvalue (of order one). This is useful because:

- It provides a direct and 'sampling-free' way, as opposed to many other fitness landscapes analysis based on sampling [82, 92, 114, 120], to know if any given elementary landscape belongs or not to the (quasi-)convex or (quasi-)concave classes simply by looking at the eigenvalue order. Methodologies to find elementary landscapes analytically [20] have been proposed already, from which the eigenvalue order would be determined as well.

Unlike related work of Moraglio [100, 101], my previous result provides the GF with a means to decide if a given combinatorial problem associated with an elementary landscape belongs in the convex, concave, quasi-convex or quasi-concave landscape classes. This addresses part of future work left by Moraglio and Sudholt [104] to develop a general runtime analysis of geometric-crossover EAs.

Abstract convex elementary landscapes being a restricted class can be an advantage compared with the whole class of elementary landscapes because the latter is already a heterogeneous class of landscapes associated with combinatorial problems of varying complexity from P to NP-complete [56, 85, 121, 136]. By contrast, elementary landscapes of eigenvalue order one, defined on binary Hamming graphs, correspond to (additive) linear pseudo-Boolean functions [136, 137], which are a 'good' example of a class of functions with similar difficulty according to Jansen [69]. This suggests (quasi-)convex or (quasi-)concave elementary landscapes, and possibly other abstract convex elementary landscapes, are of similar difficulty for certain EAs.

In fact, a general runtime analysis by Moraglio and Sudholt [104], extending the GF [100], showed certain geometric-crossover EAs in expectation can exponentially outperform random search on subclasses of quasi-concave (or quasi-convex) landscapes across problems and representations. What Moraglio and Sudholt [104] did not show, which I clarify in Chapter 11, is that quasi-convex and quasi-concave landscapes are a more restricted class than one may expect: several well-known combinatorial problems correspond to Laplacian eigenvalues of order equal or higher than two and thus, as mentioned earlier, cannot be quasi-concave nor quasi-convex. Again, being a restricted class can be an advantage, even though it makes harder to find examples other than trivial (e.g. flat landscapes).

Although the practicality of abstract convex elementary landscapes remains questionable, they offer a potential means to address two important open challenges in the GF and ELT pointed out by the previous discussion: to characterise for which combinatorial problems do geometric-crossover EAs perform well [104], and, respectively, to understand the relationship between population-based EAs and spectral properties of elementary landscapes [85].

12.2 Limitations and Future Work

This thesis focused exclusively on unifying the main theoretical foundations that support the GF and ELT. Doing so entails important limitations for this thesis:

- The unification restricts to search spaces associated with finite connected graphs. Although the GF defines search spaces on finite or continuous metric spaces, ELT limits to finite connected graphs, focusing mostly on undirected and unweighted ones. Therefore, a unified theory of the GF and ELT excludes all optimisation problems not associated with finite connected graphs, including all continuous optimisation problems.

- It follows a non-probabilistic approach to analyse search operators, whether mutation or crossover, and the behaviour of EAs by focusing only on the support or parent-offspring reachability structure of search operators. Thus the unification between the GF and ELT provides no insight on any probability distribution associated with search operators and the behaviour of EAs.

- The unification restricts to subclasses of elementary landscapes. Therefore, it excludes all fitness landscapes that are *not* elementary, also including fitness landscapes that are superpositions of different elementary landscapes such as

those associated with subset-sum or quadratic assignment problems for single-bit flip neighbourhood [20]. Particularly, this thesis focuses on characterising abstract convex elementary landscapes, rather than formulating predictive measures to estimate performance [92, 114]. Therefore, this thesis is only qualitative regarding fitness landscape analysis.

Whether a unified theory of the GF and ELT can overcome these limitations is out of this thesis's scope and thus remains an important part of future work. Nevertheless, what it is worth emphasising is that, in spite of all the limitations of this thesis, it still shows a three-fold unification between the GF and ELT concerning three critical areas: crossovers, evolutionary search and fitness landscapes. That was the purpose since the beginning of this thesis after all.

The rest of this section briefly recapitulates other relevant pieces of future work from previous chapters of this thesis.

One main avenue to continue this thesis is to further develop the crossover classification presented in Chapter 5 since it is the foundation of, and therefore affects, subsequent parts of the thesis. Chapter 5 already provides an appropriate methodology to do so, by classifying other examples of crossovers with respect to the classes of geometric crossovers and recombination P-structures, or proposing new subclasses of these. In particular, to develop the crossover classification may clarify two questions arising from Chapter 5:

- Is the intersecting-Hamming-balls crossover the only example of a highly local recombination P-structure that is not geometric? (see Section 5.5.1.)
- What examples of strict size-monotonic recombination P-structures are non-geometric crossovers? Put differently, what specific examples of non-geometric recombination P-structures fulfil all the inbreeding properties of geometric crossovers? (see Section 5.5.2.)

The answer to these questions would contribute to a more precise understanding of the differences and similarities between geometric crossovers and recombination P-structures. This is important in connection with Chapter 8 because the classes of geometric crossovers and recombination P-structures can lead to a different, possibly degenerate and uninformative, form of abstract convex evolutionary search. Characterising the relationship between crossover classes, their abstract convexity and the resulting abstract convex evolutionary search across problems and representations is an essential part of the future work in Chapter 8.

An unexplored alternative in Chapter 8 is to consider *approximated* forms of abstract convex evolutionary search as proposed by Moraglio [100]. However, this involves extending the GF to metric measurable spaces. Moreover, it is seems unclear how the crossover classification proposed in Chapter 5 would help since it ignores probability distributions. Nevertheless, this line of research is worth exploring because it is one of the challenges in the general runtime analysis of geometric-crossover EAs [104], which should be taken into account for a general unified theory of the GF and ELT.

Also, Chapter 8 proved that abstract interval convex evolutionary search can be casted as a nested inclusion chain of invariant subsets or generalised schema in the sense of Mitavskiy and He [98, 99], closely related to Vose's Boolean predicates [148] and Hofmeyr's extension [64] of Radcliffe's forma analysis [117]. This suggests a generalised schema theorem may be possible not only for geometric crossovers, as Moraglio [100] pointed out, but for recombination P-structures or more general crossovers associated with finite interval spaces like symmetric Davis's order (see Section 8.2.1), which need not be geometric.

Regarding Chapter 11, there are several lines of future research to extend abstract convex elementary landscapes. The first is to find more examples, besides uniform crossover and identity crossover, of recombination P-structures fulfilling the properties of backbone distance-transitivity and generous transitivity covered in Chapter 10 (see Section 10.3). These properties allow Chapter 11 to conceive abstract convex elementary landscapes precisely in the intersection between the class of abstract convex landscapes (Chapter 9), the class of elementary landscapes and the class of recombination P-structure elementary landscapes (Chapter 10). To find such particular class of recombination P-structures, the crossover classification can be a helpful resource.

A second major piece of future work in Chapter 11, is to find other examples of non-degenerate abstract convex elementary landscapes, hopefully more interesting than (perturbed) one-max fitness functions and for other search spaces than binary sequences. One potential example is the fitness function of the linear assignment problem for the Cayley graph induced by transpositions (or swaps) on permutations [121, 136] because it is a known example of a (Fujiyama) elementary landscape of Laplacian eigenvalue of order one [85, 121]. This is in keeping with the fact that classes of (quasi-)convex or (quasi-)concave elementary landscapes have an eigenvalue of order less than two, according to Chapter 11. For the approximated versions of abstract convex elementary landscapes it is unclear if they also follow any specific pattern in terms of their position in the Laplacian spectrum, thus remaining

as future work. Stadler [137, 139] conjectured that elementary landscapes with low-order eigenvalues are smoother than those with high order eigenvalues. I speculate that (approximately) abstract convex elementary landscapes become less globally convex for higher order eigenvalues: if present, such pattern provides a direct and analytical way to classify abstract convex elementary landscapes based on their eigenvalue order. This also aligns well with another unexplored feature of elementary landscapes suggested by Stadler [136]: the total number discrete nodal domains (see Section 10.4). They are indicative of the total number connected components of local optima of above-average fitness, which might be useful to guide evolutionary search, and can be upper-bounded depending on the eigenvalue order.

Appendix A

Computer Programs

This appendix contains the source code of computer programs that contributed to the elaboration of some examples and results in this thesis. These programs are available online at the public repository [47]. They were developed and tested in the GNU/Linux Ubuntu 16.04 LTS operating system, using Python 3.5 and Wolfram Mathematica 11.2 Student Edition. The output of these programs is omitted here.

A.1 Uniform Crossover

The following `Mathematica` notebook implements the support function of uniform crossover (Definition 3.5), and its recursive pre-hull operator, for binary Hamming sequences.

```
(* file: uniform-crossover.nb *)

(* Auxiliary functions *)

hammingSeqs[dim_] := Tuples[{0,1}, dim];
isInHammingSegment[x_,y_,z_] :=
    HammingDistance[x,z] + HammingDistance[z,y]
    == HammingDistance[x,y];
hammingSegment[x_,y_,dim_] :=
    Select[hammingSeqs[dim], ((isInHammingSegment)[x,y,#] &)];

(* Uniform crossover *)
uniform[x_,y_,dim_] := hammingSegment[x,y,dim];

(* Recursive pre-hull operator of uniform crossover
*)
closure[dim_,s_,0] :=
  s;
closure[dim_,s_,1] :=
  Union[Flatten[Union @@ Table[uniform[x,y,dim],
                               {x,s},
                               {y,s}],
                1]];
closure[dim_,s_,k_] :=
  closure[dim, closure[dim,s,1], k-1];
```

```
(* Test *)
(* example parents *)
x1 = {0,0,0};
y1 = {1,0,0};
(* offspring of x1 and y1 *)
uniform[x1,y1,3]
(* pre-hull operator on S = {x1,y1} after 2 iterations *)
closure[3, {x1,y1}, 2]
```

A.2 Symmetric Davis's Order Crossover

The following `Python` program implements the support function of Davis' order crossover (Definition 5.2) and its symmetric version (Definition 5.3), and the corresponding recursive pre-hull operator.

```
# -*- coding: utf-8 -*-
# file: davis.py

import itertools

## AUXILIARY FUNCTIONS

# remove_duplicates(ls):
#   Input:  list ls
#   Output: ls without duplicate elements
#   Note: remove_duplicates also works if ls is a list of lists.
def remove_duplicates(ls):
    # groupby needs ls to be sorted first
    new = sorted(ls)
    return list(key for key,val in groupby(new))

# flatten(ls)
#   Input: list ls
#   Output: flattens all sublists of ls
#   Example:
#       ls1 = [[1,2], [2,3]]
#       ls2 = [[1,2,3], [1,2,3], [[1,2,3],[3,2,1]]]
#       flatten(ls1) = [1,2,2,3]
#       flatten(ls2) = [1, 2, 3, 1, 2, 3, [1, 2, 3], [3, 2, 1]]
def flatten(ls):
  return list(itertools.chain.from_iterable(ls))

# ls1_setminus_ls2(ls1,ls2):
#   Input:  ls1, the list we want to remove elements from.
#   Input:  ls2, the list with elements we want to remove in ls1.
#   Output: ls1 setminus ls2, elements in ls1 not appearing in ls2.
#   Examples:
#       ls1 = [1,2,3,4], ls2 = [2,3],       out = [1,4]
#       ls1 = [2,3,9],   ls2 = [1,2,2,4,3], out = [9]
#   Note: ls1 and ls2 must not contain nested lists
#   Note: removing duplicates in ls2 beforehand is harmless to the
#   end result. All we care in ls2 is which elements appear, not
```

```
#    their order or number of occurrences.
def ls1_setminus_ls2(ls1,ls2):
    ls2_set = set(ls2)
    return [item for item in ls1 if item not in ls2]

## DAVIS ORDER CROSSOVER
#
# davis_xover(x,y,i,j):
#    Input: x, the 1st  parent used as the 'cutter' string.
#    Input: y, the 2nd parent used as the 'filler' string.
#    Input: i, the index for the start of crossover section.
#    Input: j, the index for the end of crossover section.
#    Output: the offspring z of x and y that preserves relative
#            order of symbols of the parents.
#    * x and y are lists of the same length.
#    * 1 ≤ i < j ≤ n.
#    * Main steps:
#        1.  Get fillers from y not appearing in the crossover
#            section. Consider the fillers list as a queue.
#        2.  For each index in 1..n do:
#        2.1.    If index belongs to xover section then:
#        2.1.1.      # paste x[index] to z[index]
#        2.2     Else: # outside xover section
#        2.2.1.      # pull next unused filler from y
#        2.2.2.      # paste filler to z[index]
#
def davis_xover(x,y,i,j):
    z = []
    length = len(x)
    xover_sect = x[i:(j+1)] # = [i, j+1) = [i, j]
    # Main steps
    fillers = ls1_setminus_ls2(y,xover_sect)
    for at in range(length):
        if i <= at <= j:
            z.insert(at,x[at])
        else:
            # pop returns 1st elem and deletes it.
            filler = fillers.pop(0)
            z.insert(at,filler)
    return z

# davis_xover_support(x,y):
#    Input: x, the 1st  parent used as the 'cutter' string.
#    Input: y, the 2nd parent used as the filler' string.
#    Output: all possible different offspring obtained using davis
#            order crossover on parents x and y, for all possible
#            crossover sections.
def davis_xover_support(x,y):
  offspring = []
  length = len(x)
  for i in range(0,length):
    for j in range(i,length):
      z = davis_xover(x,y,i,j)
      offspring.append(z)
  return remove_duplicates(offspring)
```

```
## SYMMETRIC DAVIS ORDER CROSSOVER
#
# symmetric_davis_xover_support(x,y):
#   Input: parent x
#   Input: parent y
#   Output: all possible different offspring obtained using davis
#           order crossover on parent pairs (x,y) and (y,x) for
#           all possible crossover sections.
def symmetric_davis_xover_support(x,y):
    offspring = []
    length = len(x)
    for i in range(0,length):
        for j in range(i,length):
            z1 = davis_xover(x,y,i,j)
            z2 = davis_xover(y,x,i,j)
            offspring.append(z1)
            offspring.append(z2)
    return remove_duplicates(offspring)

## RECURSIVE PRE-HULL OPERATOR FOR (DAVIS ORDER) CROSSOVERS
# closure(xover,sett,k):
#   Input: xover, a binary crossover function. For instance,
#          davis_xover_support or symmetric_davis_xover_support.
#   Input: sett, a set of individuals (permutations)
#   Input: k, a natural number (number of recursive calls)
#   Output: all possible descendants that can be obtained by
#           applying xover crossover to the input set sett, and
#           then again to the offspring set, and again to the
#           grandchildren, for as many times as k.
def closure(xover,sett,k):
  closed = []
  offspring = []
  if k == 0:
    closed = sett
  else:
    for x in sett:
      for y in sett:
        offspring.append(xover(x,y))
    offspring = remove_duplicates(flatten(offspring))
    closed = closure(xover,offspring,k-1)
  return closed

## MAIN
def main():
    # Parent examples
    x1 = [1,2,3] # [1,2,3,4], ['A','B','C','D']
    y1 = [3,1,2] # [2,3,4,3], ['B','A','D','C']

    offspring = symmetric_davis_xover_support(x1,y1)
    # closure of x1 and y1 after two iterations
    # closed = closure(symmetric_davis_xover_support, [x1,y1], 2)

    print("Parent x: " + str(x1))
    print("Parent y: " + str(y1))
    print("# Different offspring: " + str(len(all1)))
```

```
    print("Offspring:")
    print(all1)

if __name__ == "__main__":
    main()
```

A.3 Intersecting-Hamming-balls Crossover

The following `Mathematica` notebook implements the support function of intersecting-Hamming-balls crossover (Definition 5.5) for binary Hamming sequences, its recursive pre-hull operator, and size-monotonicity of recombination P-structures (Definition 4.3).

```
(* file: intersecting-hamming-balls.nb *)

(* Auxiliary functions with parameters
       dim: dimension
       d:   distance
       c:   center
       r:   radius
       p:   point
*)
hammingSeqs[dim_] := Tuples[{0,1}, dim];
isInHammingBall[p_,c_,r_] := HammingDistance[c,p] <= r;
hammingBall[c_,r_,dim_] :=
  Select[hammingSeqs[dim], ((isInHammingBall)[#,c,r] &)];

(* Intersecting-Hamming-balls crossover *)
hammingIntersect[x_,y_,dim_] :=
  Intersection[hammingBall[x, HammingDistance[x,y], dim],
               hammingBall[y, HammingDistance[y,x], dim]];

(* Recursive pre-hull operator of intersecting-Hamming-balls
   induced for a given set s for k iterations
*)
closure[dim_,s_,0] :=
  s;
closure[dim_,s_,1] :=
  Union[Flatten[Union @@ Table[hammingIntersect[x,y,dim],
                               {x,s},
                               {y,s}],
                1]];
closure[dim_,s_,k_] :=
  closure[dim, closure[dim,s,1], k-1];

(* Size-monotonicity property of recombination P-structures *)

(* areMonotonic outputs true if two specific pairs of
   individuals (a1,a2) and (b1,b2) verify size-monotonicity.
*)
areMonotonic[a1_,a2_,b1_,b2_,dim_] :=
  Length[hammingIntersect[a1,a2,dim]] <=
  Length[hammingIntersect[b1,b2,dim]];
```

```
(* areChildrenMonotonic outputs true if for two given parents x
   and y size-monotonicity is fulfilled: |R(x,z)| <= |R(x,y)|
*)
areChildrenMonotonic[x_,y_,dim_] :=
  ((areMonotonic[x,#,x,y,dim]) &)
    /@ hammingIntersect[x,y,dim]
(* monotonicity verifies size-monotonicity for all pairs of
   parents, that is binary Hamming sequences of dimension dim.
*)
monotonicity[dim_] :=
  And @@ Flatten[Table[areChildrenMonotonic[x,y,dim],
                       {x,hammingSeqs[dim]},
                       {y,hammingSeqs[dim]}]];
(* Test *)
(* example parents *)
x1 = {0,0,1};
y1 = {0,0,0};
(* offspring of x1 and y1 *)
hammingIntersect[x1,y1,3]
(* pre-hull operator on S = {x1,y1} after 4 iterations *)
closure[3, {x1,y1}, 4]
(* check size-monotonicity on three-dimensional binary sequences *)
monotonicity[3]
```

A.4 Hamming Ball-mutation Segment Crossover

The following Mathematica notebook implements the support function of ball-mutation segment crossover (Definition 8.2), and its recursive pre-hull operator, for binary Hamming sequences.

```
(* file: hamming-ball-mutation-segment.nb *)

(* Auxiliary functions with parameters
       dim: dimension
       d:   distance
       c:   center
       r:   radius
       p:   point
*)
hammingSeqs[dim_] := Tuples[{0,1}, dim];
isInHammingBall[p_,c_,r_] := HammingDistance[c,p] <= r;
hammingBall[c_,r_,dim_] :=
  Select[hammingSeqs[dim], ((isInHammingBall)[#,c,r] &)];
isInHammingSegment[x_,y_,z_] :=
    HammingDistance[x,z] + HammingDistance[z,y]
    == HammingDistance[x,y];
hammingSegment[x_,y_,dim_] :=
    Select[hammingSeqs[dim], ((isInHammingSegment)[x,y,#] &)];
(* Hamming ball-mutation segment crossover (any radius r) *)
hammingBallSegment[x_,y_,r_,dim_] :=
    Union[Flatten[Table[hammingBall[z,r,dim],
                        {z,hammingSegment[x,y,dim]}],
                  1]];
```

```
(* Recursive pre-hull operator of Hamming ball-mutation segment
   crossover with radius 1 induced on a given set s after k
   iterations
*)
closure[dim_,s_,0] :=
  s;
closure[dim_,s_,1] :=
  Union[Flatten[Union @@ Table[hammingBallSegment[x,y,1,dim],
                               {x,s},
                               {y,s}],
                1]];
closure[dim_,s_,k_] :=
  closure[dim, closure[dim,s,1], k-1];

(* Test *)
(* example parents *)
x1 = {0,0,0};
y1 = {1,0,0};
(* offspring of x1 and y1 *)
hammingBallSegment[x1,y1,1,3]
(* pre-hull operator on S = {x1,y1} after 2 iterations *)
closure[3, {x1,y1}, 2]
```

A.5 Random Walk

The following Mathematica notebook implements the finite Markov chain, seen in the example of Section 7.2.1, for the search space induced by the single-bit flip neighbourhood on two-dimensional binary Hamming sequences.

```
(* file: markov.nb *)

(* Finite Markov chain on the mutation search space induced by
   single-bit flip mutation on two-dimensional binary Hamming
   sequences, that is a two-dimesional hypercube graph  H(2,2)
*)

(* adjacency, diagonal and transition matrices *)
hyp22 = HypercubeGraph[2];
hyp22Adj = AdjacencyMatrix[hyp22];
hyp22Diag = DiagonalMatrix[VertexDegree[hyp22]];
hyp22Trans = hyp22Adj.Inverse[hyp22Diag];
(* markov chain:
      start1: initial probability distribution over states 1..4
      p1: markov process
*)
start1 = {1,0,0,0};
p1 = DiscreteMarkovProcess[start1,hyp22Trans];
(*    display state transition diagram         *)
Graph[p1]
(*    key properties of the markov chain           *)
MarkovProcessProperties[p1]
(*    stationary distributions for the four states *)
(PDF[StationaryDistribution[p1],#] &) /@ Range[4]
```

A.6 Headless-chicken Crossover Random Walk

The following Mathematica notebook implements the finite Markov chain, seen in the example of Section 7.3.2, for the recombination search space induced by uniform crossover on three-dimensional binary Hamming sequences.

```
(* file: markov.nb *)

(* Finite Markov chain on the recombination search space defined
   by the uniform crossover recombination P-structure on three-
   dimensional binary Hamming sequences
*)

(* generalised adjacency matrix and transition matrix *)
masks[dim_] := Tuples[{0,1}, dim];
unifCrossoverSmatrix[dim_] :=
  Table[2*((3/2)^dim)*(3^(-HammingDistance[x,y])),
        {x,masks[dim]},
        {y,masks[dim]}];
unifCrossoverTrans[dim_] :=
  (1/(2*(2^dim))) unifCrossoverSmatrix[dim];
(* verify that each row of the transition matrix adds up to 1 *)
Total[((unifCrossoverTrans[3][[#]] &) /@ Range[8])]

(* markov chain:
      start3: initial probability distribution over states 1..8
      p3: markov process
*)
start3 = {1,0,0,0,0,0,0,0};
p3 = DiscreteMarkovProcess[start3, unifCrossoverTrans[3]];
(*    display state transition diagram              *)
Graph[p3]
(*    key properties of the markov chain            *)
MarkovProcessProperties[p3]
(*    stationary distributions for the eight states *)
(PDF[StationaryDistribution[p3],#] &) /@ Range[8]
```

Glossary

Other notations and technical terms are introduced in the relevant chapters of this thesis as needed.

Sets and Multi-sets

$\mathbb{N}$
: The set of *natural numbers* $\{1, 2, 3, \ldots\}$.

$\mathbb{N}_0$
: The set of *natural numbers including zero* $\{0\} \cup \mathbb{N}$.

$\mathbb{Z}$
: The set of *whole numbers* or *integers* $\{\ldots, -1, 0, 1, \ldots\}$.

$\mathbb{R}$
: The set of *real numbers*.

$\mathcal{H}_q^n$
: The set of n-dimensional *Hamming sequences* for a q-ary alphabet with $q \geq 2$, namely the n-fold Cartesian product of $\mathcal{H}_q \stackrel{\text{def}}{=} \{0, 1, \ldots, q-1\}$.

$\mathcal{P}(X)$
: The *power set* of a set X: the set of all subsets of X including the empty set and X itself.

Y^X
: The *set of all functions* $f : X \to Y$ for arbitrary sets X and Y.

$L_{\leq c}(f) \stackrel{\text{def}}{=} \{x \in X \mid f(x) \leq c\}$
: The *sub-level set* of a function f at an image value c. Likewise, *super-level*, *strict sub-level*, *strict super-level* and *level* sets are defined by replacing the relation $\leq_{\textit{less or equal than}}$ with relations: $\geq_{\textit{greater or equal than}}$, $<_{\textit{less than}}$, $>_{\textit{greater than}}$ and $=_{\textit{equal to}}$ respectively.

Multi-set
: A *multi-set* is a pair (A, m) where A is the *underlying set* of the multi-set and $m : A \to \mathbb{N}$ is a function that indicates the *multiplicity* (i.e. number of occurrences) of each element in the multi-set.

Multi-set union
: Let (A, m_A) and (B, m_B) be two multi-sets. Their *union* is the multiset (C, m_C), such that $C = A \cup B$ and $\forall x \in C : m_C(x) = \max\{m_A(x), m_B(x)\}$.

Multi-set sum
: Let (A, m_A) and (B, m_B) be two multi-sets. Their *sum* is the multiset (C, m_C), such that $C = A \cup B$ and $\forall x \in C : m_C(x) = m_A(x) + m_B(x)$.

Graphs

$G \stackrel{\text{def}}{=} (V(G), E(G))$
: A *graph* (undirected or directed, unweighted, without loops and multiple edges), with vertex set $V(G)$ and edge set $E(G)$, respectively V and E when the graph is clear from the context.

$G_1 \simeq G_2$
: The graphs G_1 and G_2 are *isomorphic*.

$G[S]$
: The *induced subgraph* on a given graph G by a subset S of its vertices $V(G)$. $G[S]$ consists of the vertex set S and all edges in $E(G)$ with both endpoints in S.

K_n
: The *complete graph* with n vertices where any two vertices are connected by an edge.

$K_{n,m}$
: The *complete bipartite graph* with vertex set $X \cup Y$ given by disjoint sets X and Y, where $|X| = n$, $|Y| = m$, and any $\{x, y\} \in X \times Y$ is an edge.

S_n $(\simeq K_{1,n-1})$
: The *star graph* (or complete bipartite graph $K_{1,n-1}$) is a tree with n vertices, one of which having $n - 1$ neighbours and the rest of the vertices connected only to the previous one.

$H(n, q)$
: The *Hamming graph* with vertex set $\mathcal{H}_q^n$ where any two distinct sequences $x, y \in \mathcal{H}_q^n$ at Hamming distance one are connected by an edge.

$H(n, 2)$
: The *hypercube graph* with 2^n vertices.

Vectors and Matrices

$\mathbf{A} = (a_{i,j}) \in \mathbb{R}^{n \times m}$
: A $n \times m$ dimensional *matrix* $\mathbf{A}$ of real-valued entries $a_{i,j} \in \mathbb{R}$, indexed by the i-th row and the j-th column, where $1 \leq i \leq n$ and $1 \leq j \leq m$.

$\mathbf{A}^{-1}$
: The *inverse* of a matrix $\mathbf{A}$ where $\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$.

$\mathbf{A}^{\mathrm{T}}$
: The *transpose* of a matrix $\mathbf{A}$ where $(\mathbf{A}^{-1})^{\mathrm{T}} = (\mathbf{A}^{\mathrm{T}})^{-1}$.

$\mathbf{I}$
: The *identity matrix*.

$\mathbf{1}$
: The *all-ones vector* $(1, 1, \ldots, 1)$.

Probability

$X \sim P$	The random variable X is *drawn from* the probability distribution P.
Unif	The *uniform probability distribution*, whether continuous or discrete.
$\mathbb{E}[X]$	The *expectation* (or *mean value*) of a random variable X.

Optimisation

Single-objective optimisation problem $(\Omega, f, \mathcal{E}, \mathcal{I})$
: It is defined as:

$$\begin{aligned} \text{minimise} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0, \quad \forall i \in \{1, \ldots, m\} \\ \text{and} \quad & h_j(x) = 0, \quad \forall j \in \{1, \ldots, p\} \end{aligned}$$

where $x = (x_1, x_2, \ldots, x_n)$ is the optimisation variable in some domain Ω, $f(x)$ the objective function to be optimised, $g_i(x)$ the ith-inequality constraint in the set $\mathcal{I}$, and $h_j(x)$ is the jth-equality constraint in the set $\mathcal{E}$, for $i, j \in \mathbb{N}$.

Global optimum solution
: Let a single-objective optimisation problem $(\Omega, f, \mathcal{E}, \mathcal{I})$, where f is the objective function defined in the domain Ω and subject to sets of constraints $\mathcal{E}$ and $\mathcal{I}$. An optimisation variable x is a *global optimum* with respect to minimisation if $\forall y \in \Omega$ such that $y \neq x$: $f(x) \leq f(y)$, and a *strict global optimum* if $f(x) < f(y)$ holds.

Local optimum solution
: Let a single-objective optimisation problem $(\Omega, f, \mathcal{E}, \mathcal{I})$, where f is the objective function defined in a metric space Ω and subject to sets of constraints $\mathcal{E}$ and $\mathcal{I}$. An optimisation variable x is a *local optimum* with respect to minimisation if $\forall y \in N_\varepsilon(x)$ such that $y \neq x$: $f(x) \leq f(y)$, for an ε-neighbourhood $N_\varepsilon(x) \stackrel{\text{def}}{=} \{y \in \Omega \mid d(x, y) \leq \varepsilon\}$ induced by a metric d up to a distance ε. It is a *strict local optimum* if $f(x) < f(y)$ holds.

Bibliography

[1] A. A. Ahmadi, A. Olshevsky, P. A. Parrilo, and J. N. Tsitsiklis. NP-hardness of deciding convexity of quartic polynomials and related problems. *Mathematical Programming*, 137(1–2):453–476, 2013. Springer. https://doi.org/10.1007/s10107-011-0499-2. (Cited on page 229)

[2] L. Altenberg. The Schema Theorem and Price's Theorem. In D. Whitley and M. D. Vose, editors, *Foundations of Genetic Algorithms*, volume 3, pages 23–49. Elsevier, 1995. https://doi.org/10.1016/B978-1-55860-356-1.50006-6. (Cited on page 29)

[3] A. Auger and B. Doerr. *Theory of Randomized Search Heuristics: Foundations and Recent Developments*, volume 1 of *Series on Theoretical Computer Science*. World Scientific, 2011. https://doi.org/10.1142/7438. (Cited 8 times on pages 26, 29, 30, 31, 32, 34, 199, 200)

[4] T. Bäck, D. B. Fogel, and Z. Michalewicz, editors. *Handbook of Evolutionary Computation*. IOP Publishing Ltd., 1st edition, 1997. ISBN 0750303921. (Cited 15 times on pages 16, 17, 24, 25, 29, 30, 45, 73, 87, 88, 99, 109, 113, 114, 143)

[5] J. Barnes, B. Dimova, S. Dokov, and A. Solomon. The Theory of Elementary Landscapes. *Applied Mathematics Letters*, 16(3):337–343, 2003. Elsevier. https://doi.org/10.1016/S0893-9659(03)80054-X. (Cited 11 times on pages 19, 51, 113, 177, 179, 183, 189, 191, 192, 228, 239)

[6] C. Berge. *Hypergraphs: Combinatorics of Finite Sets*, volume 45 of *North-Holland Mathematical Library*. Elsevier, 1st edition, 1989. ISBN 9780080880235. (Cited 2 times on pages 44, 55)

[7] N. Biggs. Algebraic Potential Theory on Graphs. *Bulletin of the London Mathematical Society*, 29(6):641–682, 1997. London Mathematical Society. https://doi.org/10.1112/S0024609397003305. (Cited 2 times on pages 19, 220)

[8] T. Bıyıkoğlu, J. Leydold, and P. F. Stadler. *Laplacian Eigenvectors of Graphs: Perron-Frobenius and Faber-Krahn Type Theorems*, volume 1915 of *Lecture Notes in Mathematics*. Springer-Verlag, 2007. https://doi.org/10.1007/978-3-540-73510-6. (Cited 14 times on pages 19, 37, 177, 179, 180, 181, 182, 184, 191, 192, 193, 194, 218, 232)

[9] C. Blum and A. Roli. Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. *ACM Computing Surveys*, 35(3):268–308, 2003. ACM. https://doi.org/10.1145/937503.937505. (Cited 4 times on pages 16, 26, 30, 237)

[10] K. D. Boese, A. B. Kahng, and S. Muddu. A new adaptive multi-start technique for combinatorial global optimizations. *Operations Research Letters*, 16(2):101–113,

1994. Elsevier Science B.V. https://doi.org/10.1016/0167-6377(94)90065-5. (Cited 3 times on pages 165, 197, 239)

[11] E. Boros and P. L. Hammer. Pseudo-Boolean optimization. *Discrete Applied Mathematics*, 123(1–3):155–225, 2002. Elsevier Science B.V. https://doi.org/10.1016/S0166-218X(01)00341-9. (Cited 6 times on pages 28, 30, 31, 33, 165, 181)

[12] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. https://doi.org/10.1017/CBO9780511804441. (Cited 5 times on pages 34, 102, 165, 166, 167)

[13] A. E. Brouwer and W. H. Haemers. *Spectra of Graphs*. Universitext. Springer-Verlag, 1st edition, 2012. https://doi.org/10.1007/978-1-4614-1939-6. (Cited 6 times on pages 19, 177, 179, 180, 182, 204)

[14] A. E. Brouwer, A. M. Cohen, and A. Neumaier. *Distance-Regular Graphs*, volume 18 of *Ergebnisse der Mathematik und ihrer Grenzgebiete. 3. Folge / A Series of Modern Surveys in Mathematics*. Springer-Verlag, 1st edition, 1989. https://doi.org/10.1007/978-3-642-74341-2. (Cited 7 times on pages 34, 79, 82, 185, 187, 188, 219)

[15] R. E. Burkard, B. Klinz, and R. Rudolf. Perspectives of Monge properties in optimization. *Discrete Applied Mathematics*, 70(2):95–161, 1996. Elsevier Science B.V. https://doi.org/10.1016/0166-218X(95)00103-X. (Cited on page 165)

[16] M. J. Burke and T. L. Perkins. Convex and subharmonic functions on graphs. *Involve: A Journal of Mathematics*, 7(2):227–237, 2014. Mathematical Sciences Publishers. https://doi.org/10.2140/involve.2014.7.227. (Cited 2 times on pages 218, 219)

[17] M. Changat, S. Klavžar, and H. M. Mulder. The All-Paths Transit Function of a Graph. *Czechoslovak Mathematical Journal*, 51(2):349–448, 2001. Institute of Mathematics, Czech Academy of Sciences. https://doi.org/10.1023/A:1013715518448. (Cited 3 times on pages 52, 75, 77)

[18] M. Changat, P. G. Narasimha-Shenoi, F. H. Nezhad, M. Kovše, S. Mohandas, A. Ramachandran, and P. F. Stadler. Transit sets of k-point crossover operators. *AKCE International Journal of Graphs and Combinatorics*, 17(1):519–533, 2020. Taylor & Francis. https://doi.org/10.1016/j.akcej.2019.03.019. (Cited 11 times on pages 52, 57, 60, 61, 71, 72, 87, 133, 135, 236, 237)

[19] C. Chatfield. *The Analysis of Time Series: An Introduction*. Texts in Statistical Science. Chapman & Hall/CRC, 6th edition, 2003. https://doi.org/10.4324/9780203491683. (Cited 4 times on pages 29, 34, 35, 110)

[20] F. Chicano, D. Whitley, and E. Alba. A Methodology to Find the Elementary Landscape Decomposition of Combinatorial Optimization Problems. *Evolutionary Computation*, 19(4):597–637, 2011. MIT Press. https://doi.org/10.1162/EVCO_a_00039. (Cited 8 times on pages 19, 183, 187, 189, 193, 226, 240, 242)

[21] F. Chicano, F. Daolio, G. Ochoa, S. Vérel, M. Tomassini, and E. Alba. Local Optima Networks, Landscape Autocorrelation and Heuristic Search Performance. In C. A. Coello Coello, V. Cutello, K. Deb, S. Forrest, G. Nicosia, and M. Pavone, editors, *Parallel Problem Solving from Nature – PPSN XII*, volume 7492 of *Lecture Notes in Computer Science*, pages 337–347. Springer, 2012. https://doi.org/10.1007/978-3-642-32964-7_34. (Cited on page 35)

[22] F. Chicano, G. Luque, and E. Alba. Autocorrelation measures for the quadratic assignment problem. *Applied Mathematics Letters*, 25(4):698–705, 2012. Elsevier. https://doi.org/10.1016/j.aml.2011.09.053. (Cited on page 35)

[23] F. Chicano, D. Whitley, and E. Alba. Exact Computation of the Expectation Curves for Uniform Crossover. In *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*, GECCO 2012, pages 1301–1308, 2012. ACM. https://doi.org/10.1145/2330163.2330344. (Cited 5 times on pages 28, 206, 207, 229, 239)

[24] D. Corus, D.-C. Dang, A. V. Eremeev, and P. K. Lehre. Level-Based Analysis of Genetic Algorithms and Other Search Processes. *IEEE Transactions on Evolutionary Computation*, 22(5):707–719, 2017. IEEE. https://doi.org/10.1109/TEVC.2017.2753538. (Cited on page 32)

[25] R. Courant and D. Hilbert. *Methods of Mathematical Physics, Volume 1*. Interscience Publishers, Inc., New York, USA, 1st edition, 1953. ISBN 047017952X. (Cited 2 times on pages 177, 192)

[26] J. C. Culberson. Mutation-Crossover Isomorphisms and the Construction of Discriminating Functions. *Evolutionary Computation*, 2(3):279–311, 1994. MIT Press. https://doi.org/10.1162/evco.1994.2.3.279. (Cited 4 times on pages 34, 54, 57, 188)

[27] C. R. Darwin. *On the origin of species by means of natural selection, or the preservation of favoured races in the struggle for life.* John Murray, London, UK, 5th edition, 1869. (Cited 3 times on pages 16, 26, 109)

[28] S. Das, A. Abraham, and A. Konar. Particle Swarm Optimization and Differential Evolution Algorithms: Technical Analysis, Applications and Hybridization Perspectives. In Y. Liu et al., editors, *Advances of Computational Intelligence in Industrial Systems*, volume 116 of *Studies in Computational Intelligence*, pages 1–38. Springer, 2008. https://doi.org/10.1007/978-3-540-78297-1_1. (Cited on page 26)

[29] Y. Davidor. Epistasis Variance: A Viewpoint on GA-hardness. In G. J. E. Rawlins, editor, *Foundations of Genetic Algorithms*, volume 1, pages 23–35. Elsevier, 1991. https://doi.org/10.1016/B978-0-08-050684-5.50005-7. (Cited on page 35)

[30] E. B. Davies, G. M. L. Gladwell, J. Leydold, and P. F. Stadler. Discrete nodal domain theorems. *Linear Algebra and its Applications*, 336(1):51–60, 2001. Elsevier. https://doi.org/10.1016/S0024-3795(01)00313-5. (Cited 4 times on pages 191, 192, 193, 232)

[31] L. Davis. Applying Adaptive Algorithms to Epistatic Domains. In A. Joshi, editor, *Proceedings of the 9th International Joint Conference on Artificial Intelligence–Volume 1*, IJCAI'85, pages 151–162. Morgan Kaufmann Publishers Inc., 1985. ISBN 0934613028. (Cited 6 times on pages 19, 27, 45, 73, 121, 150)

[32] K. A. De Jong. Genetic Algorithms Are NOT Function Optimizers. In D. Whitley, editor, *Foundations of Genetic Algorithms*, volume 2, pages 5–17. Elsevier, 1993. https://doi.org/10.1016/B978-0-08-094832-4.50006-4. (Cited 2 times on pages 16, 29)

[33] K. A. De Jong. *Evolutionary Computation: A Unified Approach.* Complex Adaptive Systems. MIT Press, 2006. ISBN 0262041944. (Cited 3 times on pages 17, 24, 25)

[34] E. Deza and M.-M. Deza. *Dictionary of Distances.* Elsevier, 1st edition, 2006.

https://doi.org/10.1016/B978-0-444-52087-6.X5000-8. (Cited 5 times on pages 18, 34, 41, 44, 202)

[35] B. Dimova, J. W. Barnes, E. Popova, and B. Colletti. Some additional properties of elementary landscapes. *Applied Mathematics Letters*, 22(2):232–235, 2009. Elsevier. https://doi.org/10.1016/j.aml.2008.03.016. (Cited 8 times on pages 18, 37, 51, 113, 177, 179, 191, 239)

[36] B. Doerr and F. Neumann, editors. *Theory of Evolutionary Computation: Recent Developments in Discrete Optimisation*. Natural Computing Series. Springer, 1st edition, 2020. https://doi.org/10.1007/978-3-030-29414-4. (Cited 9 times on pages 24, 26, 28, 29, 30, 31, 32, 33, 237)

[37] B. Doerr, D. Johannsen, T. Kötzing, F. Neumann, and M. Theile. More effective crossover operators for the all-pairs shortest path problem. *Theoretical Computer Science*, 471:12–26, 2013. Elsevier. https://doi.org/10.1016/j.tcs.2012.10.059. (Cited on page 17)

[38] S. Droste and D. Wiesmann. Metric Based Evolutionary Algorithms. In R. Poli et al., editors, *European Conference on Genetic Programming – EuroGP 2000*, volume 1802 of *Lecture Notes in Computer Science*, pages 29–43. Springer-Verlag, 2000. https://doi.org/10.1007/978-3-540-46239-2_3. (Cited 7 times on pages 28, 89, 90, 158, 169, 231, 236)

[39] S. Droste, T. Jansen, and I. Wegener. On the analysis of the $(1 + 1)$ evolutionary algorithm. *Theoretical Computer Science*, 276(1–2):51–81, 2002. Elsevier. https://doi.org/10.1016/S0304-3975(01)00182-7. (Cited 5 times on pages 27, 31, 34, 37, 238)

[40] A. Eberhard, N. Hadjisavvas, and D. T. Luc, editors. *Generalized Convexity, Generalized Monotonicity and Applications: Proceedings of the 7th International Symposium on Generalized Convexity and Generalized Monotonicity*, volume 77 of *Nonconvex Optimization and Its Applications*. Springer US, 1st edition, 2005. https://doi.org/10.1007/b102138. (Cited 2 times on pages 165, 168)

[41] P. H. Edelman and R. E. Jamison. The Theory of Convex Geometries. *Geometriae Dedicata*, 19(3):247–270, 1985. Kluwer Academic Publishers. https://doi.org/10.1007/BF00149365. (Cited on page 105)

[42] A. E. Eiben and G. Rudolph. Theory of evolutionary computation: a bird's eye view. *Theoretical Computer Science*, 229(1–2):3–9, 1999. Elsevier. https://doi.org/10.1016/S0304-3975(99)00089-4. (Cited 2 times on pages 24, 29)

[43] R. Engelking. *Generalized Topology*. Sigma Series in Pure Mathematics. Heldermann Verlag, revised edition, 1989. ISBN 9783885380061. (Cited on page 194)

[44] M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal*, 25(4):619–633, 1975. Institute of Mathematics, Czech Academy of Sciences. http://dml.cz/dmlcz/101357. (Cited 2 times on pages 193, 194)

[45] L. J. Fogel, A. J. Owens, and M. J. Walsh. *Artificial Intelligence through Simulated Evolution*. John Wiley & Sons, New York, USA, 1966. ISBN 9780471265160. (Cited 2 times on pages 25, 26)

[46] B. Freisleben and P. Merz. A Genetic Local Search Algorithm for Solving Symmetric

and Asymmetric Traveling Salesman Problems. *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation*, pages 616–621, 1996. IEEE. https://doi.org/10.1109/ICEC.1996.542671. (Cited 4 times on pages 16, 165, 197, 239)

[47] M. D. García. PhD Thesis Code Base, GitHub repository, 25 October 2019. URL https://github.com/marcosdg/phd-thesis. Last accessed on 25 October 2019. (Cited on page 245)

[48] M. D. García and A. Moraglio. Bridging Elementary Landscapes and a Geometric Theory of Evolutionary Algorithms: First Steps. In A. Auger et al., editors, *Parallel Problem Solving from Nature – PPSN XV*, volume 11102 of *Lecture Notes in Computer Science*, pages 194–206. Springer, Cham, 2018. https://doi.org/10.1007/978-3-319-99259-4_16. (Cited 8 times on pages 13, 62, 68, 92, 196, 197, 199, 206)

[49] M. D. García and A. Moraglio. A Unifying View on Recombination Spaces and Abstract Convex Evolutionary Search. In A. Liefooghe and L. Paquete, editors, *Proceedings of the 19th European Conference on Evolutionary Computation in Combinatorial Optimisation – EvoCOP 2019*, volume 11452 of *Lecture Notes in Computer Science*, pages 179–195. Springer, Cham, 2019. https://doi.org/10.1007/978-3-030-16711-0_12. (Cited 16 times on pages 13, 62, 69, 73, 74, 75, 76, 78, 79, 80, 81, 84, 89, 120, 140, 141)

[50] H. Geiringer. On the Probability Theory of Linkage in Mendelian Heredity. *Annals of Mathematical Statistics*, 15(1):25–57, 1944. Institute of Mathematical Statistics. https://doi.org/10.1214/aoms/1177731313. (Cited 2 times on pages 119, 189)

[51] P. Gitchoff and G. P. Wagner. Recombination Induced Hypergraphs: A New Approach to Mutation-Recombination Isomorphism. *Complexity*, 2(1):37–43, 1996. John Wiley & Sons. https://doi.org/10.1002/(SICI)1099-0526(199609/10)2:1<37::AID-CPLX9>3.3.CO;2-0. (Cited 16 times on pages 19, 34, 44, 52, 53, 54, 57, 58, 59, 70, 133, 135, 153, 188, 230, 236)

[52] F. Glover and G. A. Kochenberger, editors. *Handbook of Metaheuristics*, volume 57 of *International Series in Operations Research & Management Science*. Springer, 2003. https://doi.org/10.1007/b101874. (Cited 9 times on pages 16, 18, 25, 26, 27, 30, 45, 113, 237)

[53] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989. ISBN 9780201157673. (Cited 5 times on pages 17, 27, 28, 29, 150)

[54] D. E. Goldberg. Genetic Algorithms and Walsh Functions: Part I, A Gentle Introduction. *Complex Systems*, 3(2):129–152, 1989. Complex Systems Publications. (Cited 5 times on pages 28, 34, 181, 182, 231)

[55] D. E. Goldberg. Construction of high-order deceptive functions using low-order Walsh coefficients. *Annals of Mathematics and Artificial Intelligence*, 5(1):35–48, 1992. Springer. https://doi.org/10.1007/BF01530779. (Cited 3 times on pages 34, 181, 231)

[56] L. K. Grover. Local Search and the Local Structure of NP-complete Problems. *Operations Research Letters*, 12(4):235–243, 1992. Elsevier. https://doi.org/10.1016/0167-6377(92)90049-9. (Cited 12 times on pages 19, 37, 183, 189, 191, 193, 197, 199, 200, 214, 239, 240)

[57] B. Hajek. Hitting-Time and Occupation-Time Bounds Implied by Drift Analysis with Applications. *Advances in Applied Probability*, 14(3):502–525, 1982. Applied Probability Trust. https://doi.org/10.2307/1426671. (Cited on page 32)

[58] P. R. Halmos. What Does the Spectral Theorem Say? *The American Mathematical Monthly*, 70(3):241–247, 1963. Taylor & Francis. https://doi.org/10.1080/00029890.1963.11990075. (Cited on page 185)

[59] F. Harary. *Graph Theory*. CRC Press, 1st edition, 1969. ISBN 9780201410334. (Cited 4 times on pages 34, 75, 79, 82)

[60] J. He and G. Lin. Average Convergence Rate of Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*, 20(2):316–321, 2016. IEEE. https://doi.org/10.1109/TEVC.2015.2444793. (Cited on page 30)

[61] J. He and X. Yao. Drift analysis and average time complexity of evolutionary algorithms. *Artificial Intelligence*, 127(1):57–85, 2001. Elsevier. https://doi.org/10.1016/S0004-3702(01)00058-3. (Cited on page 32)

[62] J. He, C. Reeves, C. Witt, and X. Yao. A Note on Problem Difficulty Measures in Black-Box Optimization: Classification, Realizations and Predictability. *Evolutionary Computation*, 15(4):435–443, 2007. MIT Press. https://doi.org/10.1162/evco.2007.15.4.435. (Cited on page 35)

[63] J. He, T. Chen, and X. Yao. On the Easiest and Hardest Fitness Functions. *IEEE Transactions on Evolutionary Computation*, 19(2):295–305, 2015. IEEE. https://doi.org/10.1109/TEVC.2014.2318025. (Cited 5 times on pages 27, 31, 34, 37, 238)

[64] D. Hofmeyr. On the Topology of Genetic Algorithms. In S. Kambhampati, editor, *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 582–588. AAAI Press, 2016. ISBN 9781577357704. (Cited 2 times on pages 158, 243)

[65] J. H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. Complex Adaptive Systems. MIT Press, 1992. ISBN 9780262082136. (Cited 14 times on pages 16, 25, 26, 27, 28, 29, 37, 63, 65, 102, 109, 148, 157, 181)

[66] W. Hordijk. A Measure of Landscapes. *Evolutionary Computation*, 4(4):335–360, 1996. MIT Press. https://doi.org/10.1162/evco.1996.4.4.335. (Cited 2 times on pages 34, 114)

[67] D. H. Hyers and S. M. Ulam. Approximately convex functions. *Proceedings of the American Mathematical Society*, 3(5):821–828, 1952. American Mathematical Society. https://doi.org/10.2307/2032186. (Cited on page 170)

[68] C. Igel. No Free Lunch Theorems: Limitations and Perspectives of Metaheuristics. In Y. Borenstein and A. Moraglio, editors, *Theory and Principled Methods for the Design of Metaheuristics*, Natural Computing Series, pages 1–23. Springer, 2014. https://doi.org/10.1007/978-3-642-33206-7_1. (Cited 6 times on pages 17, 27, 34, 46, 163, 239)

[69] T. Jansen. On Classifications of Fitness Functions. In L. Kallel, B. Naudts, and A. Rogers, editors, *Theoretical Aspects of Evolutionary Computing*, Natural Computing Series, pages 371–385. Springer, 2001. https://doi.org/10.1007/

978-3-662-04448-3_18. (Cited 4 times on pages 33, 35, 196, 240)

[70] T. Jansen. *Analyzing Evolutionary Algorithms: The Computer Science Perspective*. Natural Computing Series. Springer, 1st edition, 2013. https://doi.org/10.1007/978-3-642-17339-4. (Cited 7 times on pages 24, 29, 30, 31, 32, 33, 34)

[71] T. Jansen and C. Zarges. Performance analysis of randomised search heuristics operating with a fixed budget. *Theoretical Computer Science*, 545:39–58, 2014. Elsevier. https://doi.org/10.1016/j.tcs.2013.06.007. (Cited on page 31)

[72] D. B. Johnson. Efficient Algorithms for Shortest Paths in Sparse Networks. *Journal of the ACM*, 24(1):1–13, 1977. https://doi.org/10.1145/321992.321993. (Cited on page 17)

[73] T. Jones. *Evolutionary Algorithms, Fitness Landscapes and Search*. Doctoral thesis, University of New Mexico, Albuquerque, New Mexico, May 1995. (Cited 13 times on pages 20, 27, 32, 34, 44, 54, 57, 59, 114, 165, 185, 237, 238)

[74] T. Jones and S. Forrest. Fitness Distance Correlation as a Measure of Problem Difficulty for Genetic Algorithms. In L. J. Eshelman, editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 184–192, Pittsburgh, Pennsylvania, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 9781558603707. (Cited on page 35)

[75] H. Kanemitsu, M. Miyakoshi, and M. Shimbo. Properties of Unimodal and Multimodal Functions Defined by the Use of Local Minimal Value Set. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, 81(1):42–51, 1998. Scripta Technica. https://doi.org/10.1002/(SICI)1520-6440(199801)81:1<42::AID-ECJC5>3.0.CO;2-8. (Cited on page 194)

[76] R. M. Karp. Reducibility among Combinatorial Problems. In R. E. Miller, J. W. Thatcher, and J. D. Bohlinger, editors, *Complexity of Computer Computations*, IBM Research Symposia Series, pages 85–103. Springer, 1972. https://doi.org/10.1007/978-1-4684-2001-2_9. (Cited 5 times on pages 27, 28, 33, 35, 199)

[77] S. A. Kauffman. *The Origins of Order: Self-organization and Selection in Evolution*. Oxford University Press, New York, New York, USA, 1993. ISBN 9780195079517. (Cited 4 times on pages 16, 32, 34, 193)

[78] S. A. Kauffman and E. D. Weinberger. The NK model of rugged fitness landscapes and its application to maturation of the immune response. *Journal of Theoretical Biology*, 141(2):211–245, 1989. Elsevier. https://doi.org/10.1016/S0022-5193(89)80019-0. (Cited on page 34)

[79] Y. Kempner, V. E. Levit, and I. Muchnik. Quasi-Concave Functions and Greedy Algorithms. In W. Bednorz, editor, *Greedy Algorithms*, chapter 25. IntechOpen, 2008. https://doi.org/10.5772/6340. (Cited 2 times on pages 165, 168)

[80] J. Kennedy and R. Eberhart. Particle Swarm Optimisation. *International Conference on Neural Networks*, 4:1942–1948, 1995. IEEE. https://doi.org/10.1109/ICNN.1995.488968. (Cited 2 times on pages 18, 26)

[81] P. Kerschke, H. H. Hoos, F. Neumann, and H. Trautmann. Automated Algorithm Selection: Survey and Perspectives. *Evolutionary Computation*, 27(1):3–45, 2019. MIT Press. ttps://doi.org/10.1162/evco_a_00242. (Cited 2 times on pages 28, 33)

[82] A. Khulood. *Fitness landscape analysis of a class of np-hard problems*. Doctoral thesis, University of Birmingham, Birmingham, UK, June 2017. (Cited 4 times on pages 33, 34, 35, 240)

[83] S. Kirkpatrick and G. Toulouse. Configuration space analysis of travelling salesman problems. *Journal de Physique*, 46(8):1277–1292, 1985. EDP Sciences. https://doi.org/10.1051/jphys:019850046080127700. (Cited 3 times on pages 32, 197, 239)

[84] C. O. Kiselman. Subharmonic Functions on Discrete Structures. In I. Sabadini, D. C. Struppa, and D. F. Walnut, editors, *Harmonic Analysis, Signal Processing, and Complexity*, volume 238 of *Progress in Mathematics*, pages 67–80. Birkhäuser Boston, 2005. https://doi.org/10.1007/0-8176-4416-4_6. (Cited 3 times on pages 19, 219, 220)

[85] K. Klemm and P. F. Stadler. Rugged and Elementary Landscapes. In Y. Borenstein and A. Moraglio, editors, *Theory and Principled Methods for the Design of Metaheuristics*, Natural Computing Series, pages 41–61. Springer, 2014. https://doi.org/10.1007/978-3-642-33206-7_3. (Cited 19 times on pages 18, 20, 37, 121, 181, 182, 189, 190, 191, 193, 197, 214, 229, 230, 238, 239, 240, 241, 243)

[86] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Complex Adaptive Systems. MIT Press, 1992. ISBN 9780262111706. (Cited 9 times on pages 19, 26, 27, 44, 45, 73, 88, 121, 156)

[87] W. B. Langdon. Elementary Bit String Mutation Landscapes. In *Proceedings of the 11th Workshop on Foundations of Genetic Algorithms*, FOGA 2011, pages 25–42. ACM, 2011. https://doi.org/10.1145/1967654.1967658. (Cited 5 times on pages 19, 199, 206, 208, 223)

[88] J. V. Leeuwen, editor. *Algorithms and Complexity*, volume A of *Handbook of Theoretical Computer Science*. Elsevier, 1990. https://doi.org/10.1016/C2009-0-15405-3. (Cited 3 times on pages 27, 33, 35)

[89] P. K. Lehre and C. Witt. Concentrated Hitting Times of Randomized Search Heuristics with Variable Drift. In H.-K. Ahn and C.-S. Shin, editors, *Algorithms and Computation, ISAAC 2014*, volume 8889 of *Lecture Notes in Computer Science*, pages 686–697. Springer, 2014. https://doi.org/10.1007/978-3-319-13075-0_54. (Cited on page 32)

[90] L. Lovász. Submodular functions and convexity. In A. Bachem, B. Korte, and M. Grötschel, editors, *Mathematical Programming The State of the Art*, pages 235–257. Springer-Verlag, 1983. https://doi.org/10.1007/978-3-642-68874-4. (Cited 2 times on pages 165, 168)

[91] G. Lu, R. Bahsoon, and X. Yao. Applying Elementary Landscape Analysis to Search-Based Software Engineering. In *Second International Symposium on Search Based Software Engineering*, pages 3–8. IEEE, 2010. https://doi.org/10.1109/SSBSE.2010.10. (Cited 3 times on pages 18, 37, 197)

[92] K. M. Malan and A. P. Engelbrecht. A Survey of Techniques for Characterising Fitness Landscapes and Some Possible Ways Forward. *Information Sciences*, 241: 148–163, 2013. Elsevier. https://doi.org/10.1016/j.ins.2013.04.015. (Cited 4 times on pages 34, 35, 240, 242)

[93] J. McDermott. When and Why Metaheuristics Researchers can Ignore "No Free

Lunch" Theorems. *SN Computer Science*, 1(60), 2020. Springer. https://doi.org/10.1007/s42979-020-0063-3. (Cited 3 times on pages 28, 34, 236)

[94] B. Mendelson. *Introduction to Topology*. Blackie & Son Ltd., Glasgow, UK, 1st edition, 1963. (Cited 2 times on pages 40, 132)

[95] A. Menon, editor. *Frontiers of Evolutionary Computation*, volume 11 of *Genetic Algorithms and Evolutionary Computation*. Springer US, 2004. https://doi.org/10.1007/b116128. (Cited 3 times on pages 17, 24, 29)

[96] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, 3rd edition, 1996. https://doi.org/10.1007/978-3-662-03315-9. (Cited 4 times on pages 17, 25, 27, 87)

[97] Z. Michalewicz. Quo Vadis, Evolutionary Computation? On a Growing Gap between Theory and Practice. In J. Liu et al., editors, *Advances in Computational Intelligence: IEEE World Congress on Computational Intelligence, WCCI 2012*, volume 7311 of *Lecture Notes in Computer Science*, pages 98–121. Springer, 2012. https://doi.org/10.1007/978-3-642-30687-7_6. (Cited 3 times on pages 17, 24, 28)

[98] B. Mitavskiy. Crossover Invariant Subsets of the Search Space for Evolutionary Algorithms. *Evolutionary Computation*, 12(1):19–46, 2004. MIT Press. https://doi.org/10.1162/evco.2004.12.1.19. (Cited 13 times on pages 28, 29, 37, 65, 103, 122, 133, 149, 157, 158, 210, 212, 243)

[99] B. Mitavskiy and J. He. Combining Drift Analysis and Generalized Schema Theory to Design Efficient Hybrid and/or Mixed Strategy EAs. *IEEE Congress on Evolutionary Computation*, pages 2028–2036, 2013. IEEE. https://doi.org/10.1109/CEC.2013.6557808. (Cited 9 times on pages 28, 29, 31, 37, 65, 122, 149, 157, 243)

[100] A. Moraglio. *Towards a Geometric Unification of Evolutionary Algorithms*. Doctoral thesis, University of Essex, Essex, UK, November 2007. (Cited 39 times on pages 18, 19, 20, 24, 36, 37, 40, 42, 43, 44, 45, 46, 47, 59, 62, 63, 70, 71, 82, 88, 90, 100, 101, 105, 106, 107, 108, 153, 163, 196, 206, 230, 235, 236, 237, 239, 240, 241, 243)

[101] A. Moraglio. Abstract Convex Evolutionary Search. In *Proceedings of the 11th Workshop on Foundations of Genetic Algorithms*, FOGA 2011, pages 151–162. ACM, 2011. https://doi.org/10.1145/1967654.1967668. (Cited 33 times on pages 18, 20, 37, 42, 47, 88, 100, 101, 105, 106, 107, 108, 120, 139, 153, 155, 164, 165, 169, 170, 172, 173, 174, 175, 176, 196, 219, 228, 230, 237, 238, 239, 240)

[102] A. Moraglio. Convex Evolutionary Search, GitHub repository, 28 September 2015. URL https://github.com/amoraglio/CES. Last accessed on 22 October 2019. (Cited on page 101)

[103] A. Moraglio and R. Poli. Inbreeding Properties of Geometric Crossover and Non-geometric Recombinations. In C. R. Stephens, M. Toussaint, D. Whitley, and P. F. Stadler, editors, *Proceedings of the 9th International Conference on Foundations of Genetic Algorithms*, FOGA 2007, pages 1–14. Springer, Heidelberg, 2007. https://doi.org/10.1007/978-3-540-73482-6_1. (Cited 10 times on pages 19, 44, 46, 47, 48, 63, 73, 84, 85, 236)

[104] A. Moraglio and D. Sudholt. Principled Design and Runtime Analysis of Abstract Convex Evolutionary Search. *Evolutionary Computation*, 25(2):205–236, 2017. MIT Press. https://doi.org/10.1162/EVCO_a_00169. (Cited 24 times on pages 18, 20,

37, 47, 62, 101, 108, 120, 157, 164, 165, 174, 175, 176, 196, 199, 229, 231, 236, 238, 239, 240, 241, 243)

[105] A. Moraglio, C. D. Chio, and R. Poli. Geometric Particle Swarm Optimisation. In R. Poli et al., editors, *European Conference on Genetic Programming – EuroGP 2007*, volume 4445 of *Lecture Notes in Computer Science*, pages 125–136. Springer, 2007. https://doi.org/10.1007/978-3-540-71605-1_12. (Cited on page 18)

[106] H. Mühlenbein and G. Paaß. From recombination of genes to the estimation of distributions I. Binary parameters. In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature – PPSN IV*, volume 1141 of *Lecture Notes in Computer Science*, pages 178–187. Springer, 1996. https://doi.org/10.1007/3-540-61723-X_982. (Cited on page 26)

[107] H. M. Mulder. Transit Functions on Graphs (and Posets). In M. Changat, S. Klavžar, H. M. Mulder, and A. Vijayakumar, editors, *Convexity in discrete structures*, volume 5 of *Ramanujan Lecture Notes Series*, pages 117–130. International Press, 2010. ISBN 9781571461902. (Cited 5 times on pages 37, 52, 61, 69, 87)

[108] S. Nallaperuma, F. Neumann, and D. Sudholt. Expected Fitness Gains of Randomized Search Heuristics for the Traveling Salesperson Problem. *Evolutionary Computation*, 25(4):673–705, 2017. MIT Press. https://doi.org/10.1162/evco_a_00199. (Cited on page 31)

[109] G. Ochoa and N. Veerapen. Deconstructing the Big Valley Search Space Hypothesis. In F. Chicano, B. Hu, and P. García-Sánchez, editors, *Proceedings of the 16th European Conference on Evolutionary Computation in Combinatorial Optimisation – EvoCOP 2016*, volume 9595 of *Lecture Notes in Computer Science*, pages 58–73. Springer, Cham, 2016. https://doi.org/10.1007/978-3-319-30698-8_5. (Cited 4 times on pages 35, 165, 232, 239)

[110] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill Series in Electrical Engineering. McGraw-Hill, 3rd edition, 1993. ISBN 9780070484771. (Cited 3 times on pages 29, 110, 112)

[111] M. Parnas, D. Ron, and R. Rubinfeld. On Testing Convexity and Submodularity. In J. D. P. Rolim and S. Vadhan, editors, *Randomization and Approximation Techniques in Computer Science – RANDOM 2002*, volume 2483 of *Lecture Notes in Computer Science*, pages 11–25. Springer, 2002. https://doi.org/10.1007/3-540-45726-7_2. (Cited 2 times on pages 165, 229)

[112] I. M. Pelayo. *Geodesic Convexity in Graphs*. SpringerBriefs in Mathematics. Springer, New York, 1st edition, 2013. https://doi.org/10.1007/978-1-4614-8699-2. (Cited 6 times on pages 37, 52, 69, 105, 132, 133)

[113] R. C. Penney. *Linear Algebra: Ideas and Applications*. John Wiley & Sons, Inc., 4th edition, 2015. ISBN 9781118909584. (Cited 3 times on pages 177, 185, 223)

[114] E. Pitzer and M. Affenzeller. A Comprehensive Survey on Fitness Landscape Analysis. In J. Fodor, R. Klempous, and C. P. S. Araujo, editors, *Recent Advances in Intelligent Engineering Systems*, volume 378 of *Studies in Computational Intelligence*, pages 161–191. Springer, 2012. https://doi.org/10.1007/978-3-642-23229-9_8. (Cited 5 times on pages 33, 34, 35, 240, 242)

[115] R. Poli. Hyperschema Theory for GP with One-Point Crossover, Building Blocks, and Some New Results in GA Theory. In R. Poli et al., editors, *European Conference*

on Genetic Programming – EuroGP 2000, volume 1802 of *Lecture Notes in Computer Science*, pages 163–180. Springer-Verlag, 2000. https://doi.org/10.1007/978-3-540-46239-2_12. (Cited on page 29)

[116] R. Poli, L. Vanneschi, W. B. Langdon, and N. F. McPhee. Theoretical results in genetic programming: the next ten years? *Genetic Programming and Evolvable Machines*, 11(3–4):285–320, 2010. Springer. https://doi.org/10.1007/s10710-010-9110-5. (Cited 4 times on pages 27, 29, 37, 238)

[117] N. J. Radcliffe. The algebra of genetic algorithms. *Annals of Mathematics and Artificial Intelligence*, 10(4):339–384, 1994. Kluwer Academic Publishers. https://doi.org/10.1007/BF01531276. (Cited 7 times on pages 27, 28, 29, 65, 133, 158, 243)

[118] I. Rechenberg. *Evolutionsstrategie—Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Dissertation, Technische Universität, Berlin, Germany, 1971. (Cited 2 times on pages 25, 26)

[119] J. R. Rice. The Algorithm Selection Problem. *Advances in Computers*, 15(1):65–118, 1976. Academic Press. https://doi.org/10.1016/S0065-2458(08)60520-3. (Cited on page 28)

[120] H. Richter and A. Engelbrecht, editors. *Recent Advances in the Theory and Application of Fitness Landscapes*, volume 6 of *Emergence, Complexity and Computation*. Springer-Verlag, 1st edition, 2014. https://doi.org/10.1007/978-3-642-41888-4. (Cited 7 times on pages 17, 24, 32, 33, 34, 35, 240)

[121] D. Rockmore, P. Kostelec, W. Hordijk, and P. F. Stadler. Fast Fourier Transforms for Fitness Landscapes. *Applied and Computational Harmonic Analysis*, 12(1):57–76, 2002. Elsevier. https://doi.org/10.1006/acha.2001.0346. (Cited 5 times on pages 18, 37, 230, 240, 243)

[122] F. J. Rodriguez, C. García-Martínez, and M. Lozano. Hybrid Metaheuristics Based on Evolutionary Algorithms and Simulated Annealing: Taxonomy, Comparison, and Synergy Test. *IEEE Transactions on Evolutionary Computation*, 16(6):787–800, 2012. IEEE. https://doi.org/10.1109/TEVC.2012.2182773. (Cited 3 times on pages 16, 26, 237)

[123] F. Rothlauf. *Representations for Genetic and Evolutionary Algorithms*. Springer-Verlag, 2nd edition, 2006. https://doi.org/10.1007/3-540-32444-5. (Cited 6 times on pages 27, 28, 34, 42, 89, 231)

[124] F. Rothlauf. *Design of Modern Heuristics: Principles and Application*. Natural Computing Series. Springer, 1st edition, 2011. https://doi.org/10.1007/978-3-540-72962-4. (Cited 9 times on pages 27, 28, 34, 89, 90, 158, 181, 231, 236)

[125] J. E. Rowe, M. D. Vose, and A. H. Wright. Structural Search Spaces and Genetic Operators. *Evolutionary Computation*, 12(4):461–493, 2004. MIT Press. https://doi.org/10.1162/1063656043138941. (Cited 4 times on pages 28, 29, 181, 187)

[126] J. E. Rowe, M. D. Vose, and A. H. Wright. Representation Invariant Genetic Operators. *Evolutionary Computation*, 18(4):635–660, 2010. MIT Press. https://doi.org/10.1162/EVCO_a_00007. (Cited on page 28)

[127] G. Rudolph. Convergence Analysis of Canonical Genetic Algorithms. *IEEE Transactions on Neural Networks*, 5(1):96–101, 1994. IEEE. https://doi.org/10.1109/

72.265964. (Cited on page 30)

[128] G. Rudolph. Convergence of Evolutionary Algorithms in General Search Spaces. In *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation*, ICEC'96, pages 50–54. IEEE, 1996. https://doi.org/10.1109/ICEC.1996.542332. (Cited on page 30)

[129] F. Schmitt and F. Rothlauf. On the Importance of the Second Largest Eigenvalue on the Convergence Rate of Genetic Algorithms. In *Proceedings of the Third Annual Conference on Genetic and Evolutionary Computation*, GECCO 2001, pages 559–564. ACM, 2001. https://dl.acm.org/doi/10.5555/2955239.2955324. (Cited on page 30)

[130] H.-P. Schwefel. *Evolutionsstrategie und numerische Optimierung*. Dissertation, Technische Universität, Berlin, Germany, 1975. (Cited 2 times on pages 25, 26)

[131] M. Shpak and G. P. Wagner. Asymmetry of Configuration Space Induced by Unequal Crossover: Implications for a Mathematical Theory of Evolutionary Innovation. *Artificial Life*, 6(1):25–43, 2000. MIT Press. https://doi.org/10.1162/106454600568302. (Cited 12 times on pages 44, 60, 70, 121, 123, 125, 126, 127, 133, 156, 230, 236)

[132] I. Singer. *Abstract Convex Analysis*, volume 25 of *Wiley-Interscience and Canadian Mathematics Series of Monographs and Texts*. John Wiley & Sons, Inc., 1997. ISBN 9780471160151. (Cited 4 times on pages 34, 165, 168, 172)

[133] V. P. Soltan and P. S. Soltan. d-convex functions. *Doklady Akademii Nauk SSSR*, 249(3):555–558, 1979. Academy of Sciences of the Moldavian SSR. http://mi.mathnet.ru/eng/dan43163. (Cited 5 times on pages 168, 170, 172, 173, 174)

[134] B. M. R. Stadler, P. F. Stadler, G. P. Wagner, and W. Fontana. The Topology of the Possible: Formal Spaces Underlying Patterns of Evolutionary Change. *Journal of Theoretical Biology*, 213(2):241–274, 2001. Elsevier. https://doi.org/10.1006/jtbi.2001.2423. (Cited 2 times on pages 60, 132)

[135] B. M. R. Stadler, P. F. Stadler, M. Shpak, and G. P. Wagner. Recombination Spaces, Metrics and Pretopologies. *Zeitschrift für Physikalische Chemie*, 216(2):217–234, 2002. De Gruyter. https://doi.org/10.1524/zpch.2002.216.2.217. (Cited 5 times on pages 44, 121, 123, 126, 132)

[136] P. F. Stadler. Towards a Theory of Landscapes. In R. López-Peña et al., editors, *Complex Systems and Binary Networks*, volume 461 of *Lecture Notes in Physics*, pages 78–163. Springer, 1995. https://doi.org/10.1007/BFb0103571. (Cited 34 times on pages 18, 19, 20, 33, 35, 37, 49, 51, 63, 109, 110, 113, 177, 179, 180, 182, 183, 184, 189, 190, 191, 193, 194, 197, 199, 228, 230, 232, 235, 238, 239, 240, 243, 244)

[137] P. F. Stadler. Landscapes and their correlation functions. *Journal of Mathematical Chemistry*, 20(1):1–45, 1996. Springer. https://doi.org/10.1007/BF01165154. (Cited 21 times on pages 18, 35, 110, 177, 179, 180, 181, 182, 183, 184, 189, 190, 191, 193, 194, 199, 204, 226, 228, 240, 244)

[138] P. F. Stadler and C. R. Stephens. Landscapes and Effective Fitness. *Comments on Theoretical Biology*, 8(4–5):389–431, 2003. ISSN 0894-8550. Taylor and Francis Ltd. https://doi.org/10.1080/08948550302439. (Cited 5 times on pages 29, 109, 177, 197, 228)

[139] P. F. Stadler and G. P. Wagner. Algebraic Theory of Recombination Spaces. *Evolu-*

tionary Computation, 5(3):241–275, 1998. MIT Press. https://doi.org/10.1162/evco.1997.5.3.241. (Cited 41 times on pages 19, 20, 34, 37, 44, 51, 52, 53, 54, 55, 57, 58, 59, 63, 70, 72, 83, 87, 113, 115, 116, 117, 153, 181, 184, 185, 186, 187, 188, 189, 190, 194, 197, 202, 228, 230, 231, 236, 238, 239, 244)

[140] P. F. Stadler, G. P. Wagner, and R. Seitz. Population Dependent Fourier Decomposition of Fitness Landscapes over Recombination Spaces: Evolvability of Complex Characters. *Bulletin of Mathematical Biology*, 62(3):399–428, 2000. Springer-Verlag. https://doi.org/10.1006/bulm.1999.0167. (Cited 20 times on pages 19, 20, 51, 63, 87, 109, 113, 115, 118, 119, 177, 184, 185, 188, 189, 197, 230, 237, 238, 239)

[141] C. R. Stephens and A. Zamora. EC Theory: A Unified Viewpoint. In E. Cantú-Paz et al., editors, *Proceedings of the 2003 Annual Conference on Genetic and Evolutionary Computation*, volume 2724 of *Lecture Notes in Computer Science*, pages 1394–1405. Springer, 2003. https://doi.org/10.1007/3-540-45110-2_13. (Cited 5 times on pages 17, 24, 29, 62, 153)

[142] R. Storn and K. Price. Differential Evolution–A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11: 341–359, 1997. Springer. https://doi.org/10.1023/A:1008202821328. (Cited on page 26)

[143] P. D. Surry and N. J. Radcliffe. Formal Search Algorithms + Problem Characterisations = Executable Search Strategies. In Y. Borenstein and A. Moraglio, editors, *Theory and Principled Methods for the Design of Metaheuristics*, Natural Computing Series, pages 247–270. Springer, 2014. https://doi.org/10.1007/978-3-642-33206-7_11. (Cited 2 times on pages 27, 28)

[144] A. Thorhauer and F. Rothlauf. On the Locality of Standard Search Operators in Grammatical Evolution. In T. Bartz-Beielstein, J. Branke, B. Filipič, and J. Smith, editors, *Parallel Problem Solving from Nature – PPSN XIII*, volume 8672 of *Lecture Notes in Computer Science*, pages 465–475. Springer, Cham, 2014. https://doi.org/10.1007/978-3-319-10762-2_46. (Cited 7 times on pages 28, 63, 84, 88, 89, 90, 158)

[145] T. Tlusty. A relation between the multiplicity of the second eigenvalue of a graph Laplacian, Courant's nodal line theorem and the substantial dimension of tight polyhedral surfaces. *Electronic Journal of Algebra*, 26:315–324, 2007. International Linear Algebra Society. https://doi.org/10.13001/1081-3810.1204. (Cited 4 times on pages 194, 204, 218, 229)

[146] M. L. J. van de Vel. *Theory of Convex Structures*. North-Holland Mathematical Library. North-Holland, 1st edition, 1993. ISBN 9780444815057. (Cited 19 times on pages 37, 41, 42, 52, 65, 69, 87, 101, 103, 104, 105, 121, 122, 131, 132, 133, 169, 236, 237)

[147] V. K. Vassilev, T. C. Fogarty, and J. F. Miller. Information Characteristics and the Structure of Landscapes. *Evolutionary Computation*, 8(1):31–60, 2000. MIT Press. https://doi.org/10.1162/106365600568095. (Cited on page 34)

[148] M. D. Vose. Generalizing the notion of schema in genetic algorithms. *Artificial Intelligence*, 50(3):385–396, 1991. Elsevier Science B.V. https://doi.org/10.1016/0004-3702(91)90019-G. (Cited 3 times on pages 29, 157, 243)

[149] M. D. Vose. *The Simple Genetic Algorithm: Foundations and Theory*. Complex Adaptive Systems. MIT Press, 1999. ISBN 9780262220583. (Cited on page 30)

[150] G. P. Wagner and P. F. Stadler. Complex Adaptations and the Structure of Recombination Spaces. In C. L. Nehaniv and M. Ito, editors, *Algebraic Engineering*, pages 96–115. World Scientific, 1999. https://doi.org/10.1142/3953. (Cited 17 times on pages 19, 20, 51, 109, 113, 177, 181, 184, 185, 188, 189, 202, 230, 236, 237, 238, 239)

[151] J. L. Walsh. A Closed Set of Normal Orthogonal Functions. *American Journal of Mathematics*, 45(1):5–24, 1923. Johns Hopkins University Press. https://doi.org/10.2307/2387224. (Cited 2 times on pages 28, 181)

[152] E. D. Weinberger. Correlated and Uncorrelated Fitness Landscapes and How to Tell the Difference. *Biological Cybernetics*, 63(15):325–336, 1990. Springer. https://doi.org/10.1007/BF00202749. (Cited 3 times on pages 32, 34, 35)

[153] D. Whitley. A Grey Box Manifesto for Evolutionary Computation. *SIGEVOlution*, 12(1):3–5, 2019. ACM. https://doi.org/10.1145/3328473.3328474. (Cited 2 times on pages 27, 33)

[154] D. Whitley. Next Generation Genetic Algorithms: A User's Guide and Tutorial. In M. Gendreau and J.-Y. Potvin, editors, *Handbook of Metaheuristics*, volume 272 of *International Series in Operations Research & Management Science*, pages 245–274. Springer, 2019. https://doi.org/10.1007/978-3-319-91086-4_8. (Cited on page 29)

[155] D. H. Wolpert and W. G. Macready. No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997. IEEE. https://doi.org/10.1109/4235.585893. (Cited 8 times on pages 17, 27, 34, 46, 88, 163, 236, 239)

[156] S. Wright. The roles of mutation, inbreeding, crossbreeding, and selection in evolution. In D. F. Jones, editor, *International Proceedings of the Sixth International Congress on Genetics*, volume 1, pages 356–366, Ithaca, New York, USA, 1932. Brooklyn Botanic Garden. (Cited on page 32)

[157] S. Wrigth. Surfaces of Selective Values Revisited. *The American Naturalist*, 131(1):115–123, 1988. The American Society of Naturalists. https://doi.org/10.1086/284777. (Cited on page 32)