Abstract Convex Evolutionary Search

Alberto Moraglio
School of Computing and Centre for Reasoning
University of Kent, Canterbury, UK
A.Moraglio@kent.ac.uk

ABSTRACT

Geometric crossover is a formal class of crossovers which includes many well-known recombination operators across representations. In this paper, we present a general result showing that all evolutionary algorithms using geometric crossover with no mutation perform the same form of convex search regardless of the underlying representation, the specific selection mechanism, the specific offspring distribution, the specific search space, and the problem at hand. We then start investigating a few representation/space-independent geometric conditions on the fitness landscape – various forms of generalized concavity – that when matched with the convex evolutionary search guarantee, to different extents, improvement of offspring over parents for any choice of parents. This is a first step towards showing that the convexity relation between search and landscape may play an important role towards explaining the performance of evolutionary algorithms in a general setting across representations.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search; G.1.6 [**Numerical Analysis**]: Optimization; F.2 [**Analysis of Algorithms and Problem Complexity**]

General Terms

Theory

1. INTRODUCTION

In the research community there is a strong feeling that the Evolutionary Computation (EC) field needs unification and systematization in a rational framework to survive its own success (De Jong [4]).

The various flavors of evolutionary algorithms (EAs) look very similar when cleared of algorithmically irrelevant differences such as domain of application, phenotype interpretation and representation-independent algorithmic characteristics that, in effect, can be freely exchanged between algorithms, such as the selection scheme. Ultimately, the origin of the differences of the various flavors of evolutionary algorithms is rooted in the solution representation and relative genetic operators.

Are these differences only superficial? Is there a deeper unity encompassing all evolutionary algorithms beyond the specific representation? Formally, is a general mathematical framework that unifies search operators for all solution representations possible at all? Would such a general framework be able to capture essential properties encompassing all EAs or would it be too abstract to say anything useful? These are important, difficult open research questions which the present paper attempts to start attacking.

A number of researchers have been pursuing EC unification across representations. Although, so far, no one has been able to build a fully-fledged theory of representations. For example, Radcliffe pioneered a unified theory of representations [12], although he never used the word "unification"; Poli unified the schema theorem for traditional genetic algorithms and genetic programming [6]; Stephens suggested that all evolutionary algorithms can be unified using the language of dynamical systems and coarse graining [18]; while Rothlauf initiated a less formal theory of representations [14]; Rowe et al., building upon Radcliffe's work, have devised a theory of representation based on group theory [15]; Stadler et al. built a theory of fitness landscapes that connects with representations and search operators [13].

In the last decade, EC theory has experienced important progress. However, the lack of a unified formal framework encompassing different solution representations is at the origin of the fragmentation of evolutionary computation theory, which has led to the development of significantly different theories for different representations and for different problems. This fragmentation is symptomatic of the fact that the very fundamental working principles underlying all evolutionary algorithms are not yet well understood. More fundamentally, the lack of a uniform formal language encompassing all representations prevents us from investigating whether such common principles exist at all.

Recent research [9] has shown that a common geometric framework is possible and that most of mutation and recombination operators across representations admit surprisingly simple common geometric characterizations, termed geometric mutation and geometric crossover, based on an axiomatic notion of distance. The geometric view of search operators formalizes and simplifies the relationship between representations, search operators and associated distance,

FOGA'11, January 5–9, 2011, A-6867 Schwarzenberg, Austria.

and it equates their induced search space and fitness landscape to the traditional notions of neighbourhood space and fitness landscape.

In this paper, we start building a representation-independent theory of evolutionary algorithms starting from the definition of geometric crossover [1]. The methodology used is based on *mathematical abstraction* (see for example [8][1]): it voluntarily ignores representation/search space specific properties of geometric crossover, and uses only those properties of geometric crossover which derive from the distance axioms, which are therefore common to all geometric crossovers across representations. Abstraction results in unification as a theory is obtained that applies to any representation and search space.

Abstraction is the key to showing that all evolutionary algorithms present a common behavioral core. In this paper, we will show that they all do the same type of convex search [2]. Naturally, those properties that a geometric crossover has by virtue of the nature of the specific class of underlying representation and search space are not within the scope of a theory of abstract search. Those properties are reserved for special investigation. To investigate such specific properties systematically within a general framework, a possible scenario consists of creating a hierarchy of less and less abstract spaces (a taxonomy of metric spaces) organized according to those characteristics that allow us to prove stronger and stronger statements on the performance of an evolutionary algorithm when specified to them. As convexity will appear to be a key element in evolutionary search, fundamental properties to consider may be the so called convexity numbers [20](e.g., Radon number and Helly number) that characterize at a more finely-grained scale the specific convexity of the underlying spaces. These numbers then would appear as parameters of a general relation characterizing the performance of an evolutionary algorithm as a function of the characteristics of the underlying space. This approach to a general theory of evolutionary algorithms follows, in spirit, the philosophy proposed by Stephens and Zamora [18] based on the notions of universality and taxonomy but it operates at a higher level of abstraction.

The convex search result is significant as it shows that indeed there is a common behavioral identity of all evolutionary algorithms that goes beyond the underlying representation. However, this result *per se* does not show that a meaningful general theory of evolutionary algorithms may be possible. Indeed, the NFL theorem [21] implies that a search algorithm must be well-matched with a certain class of fitness landscapes respecting some conditions to perform on average better than random search. As a consequence, any non-futile theory which aims at proving performance better than random search of a class of search algorithms needs to indicate with respect to what class of fitness landscapes. Therefore, an important question is: are there general conditions on the fitness landscape that guarantee good performances of the convex search for any space/representation?

Interestingly, the abstract convex search of an evolutionary algorithm suggests representation/space independent geometric conditions on the fitness landscape – various forms of generalized concavity – that guarantee to various extents that offspring improve over parents for any choice of the parents. This is an important property that links the topography of the fitness landscape with the parent/offspring fitness heritability throughout the evolutionary process. This shows that the underlying convexity relation between search and landscape *per se* may play a key role towards explaining the performance of evolutionary algorithms in a representation-independent fashion. In this paper, we consider a number of generalizations of the notion of concave function and approximately concave function to general metric spaces and start investigating their suitability as classes of fitness landscapes to employ as a basis for a theory of abstract convex evolutionary search.

In the long term, this theoretical framework may have interesting links with the theory of convex optimization [3], in which the notion of convexity of sets and functions is central. Most of the results in convex optimization pertain to continuous optimization, but there is ongoing research aimed at generalizing the results for continuous spaces to discrete spaces [11]. There are, however, at least two important differences between the generalized notion of convexity in convex optimization and that of the present paper: (i) in convex optimization, the discrete spaces considered for the generalization are restricted to spaces of integer vectors, rather than being general metric spaces encompassing, as important special case, combinatorial spaces based on structured representations (e.g., trees) as, instead, it is intended in the present work; (ii) in convex optimization, the generalization of convex function focuses on preserving and exploiting the property of traditional convex function of being unimodal, so that local and global optima always coincide. Instead the emphasis of the present work is on generalizing the notion of convex trend in the attempt to provide a formalization of the well-known notion of global convexity of the fitness landscape [2] that is known experimentally to be beneficial for the performance of evolutionary algorithms. This would make a theory based on the framework started in this paper of practical relevance because fitness landscapes normally associated with many important combinatorial problems have been shown to be globally convex [7].

It is worth mentioning that the results presented in this paper may look deceptively simple at first. On one hand, they are perfectly aligned with the geometric intuition everyone has about the Euclidean space. On the other hand, they are very general as they apply to general metric spaces and across representations. The latter aspect is non-trivial as only very few properties of the Euclidean space hold for general metric spaces. Many other properties break down, often in unexpected ways, in the transition from specific to more general spaces. An important contribution of this paper is to present results that derive only from those intuitive properties of the Euclidean space that hold for general metric spaces, so that their geometric intuition can be retained in the general context. This is insightful and allows us, for example, to apply the same geometric reasoning on continuous spaces and combinatorial spaces, even if in many respects they are very different types of spaces. The chal-

[1]In this paper, we do not consider mutation, not because we consider it an unimportant operator, but rather because the dynamics of the resulting search cannot be described purely in geometric terms. Mutation requires a more complex theoretical framework which combines abstract geometry with measure-theoretic elements. We leave this as future work.

[2]This property might be the only one all evolutionary algorithms with geometric crossover have in common. Nonetheless, this is an important property as it may be central to explain their performance in a unified way.

lenge in this line of theory is choosing appropriate general definitions that shape the framework and that allow us to generalize theorems holding for the Euclidean space in a natural and straightforward way.

2. ABSTRACT CONVEXITY

In this section we introduce two notions of abstract convexity, which are obtained by generalising the traditional notion of convex set in different directions, and show how they are related. Both notions of convexity and their relations are necessary to prove the results in the subsequent sections.

2.1 Preliminaries: Balls and Segments

A metric is a generalization of the notion of distance. A metric space is a set X with a distance function d (the metric) that, for every two points x and y in X, gives the distance between them as a nonnegative real number $d(x, y)$. A metric space must also satisfy:

1. $d(x, y) = 0$ iff $x = y$
2. $d(x, y) = d(y, x)$
3. $d(x, z) + d(z, y) \geq d(x, y)$ for all z in X

Given a metric space $M = (X, d)$ the *line segment* between x and y, termed extremes, is the set $[x, y]_d = \{z \in X | d(x, z) + d(z, y) = d(x, y)\}$, and the *closed ball* is the set $B_d(x; r) = \{y \in S | d(x, y) \leq r\}$ where r is a positive real number called the *radius* of the ball. Examples of balls and segments for different spaces are shown in Figure 1. Note how the same set can have different geometries (see Euclidean and Manhattan spaces) and how segments can have more than one pair of extremes. For instance, in the Hamming space, a segment coincides with a hypercube and the number of extremes varies with the length of the segment, while in the Manhattan space, a segment is a rectangle and it has two pairs of extremes. Also, a segment is not necessarily "slim", that is, it may include points that are not on the boundaries. Furthermore, a segment does not coincide with a shortest path connecting its extremes (*geodesic*). In general, there may be more than one geodesic connecting two extremes.

2.2 Definition of Abstract Convexity

In the Euclidean space, a set is convex iff the line segment connecting any two points in the set lies entirely in the set. A natural way of generalizing the notion of convex set to more general spaces is to generalize the notion of line segment and define convex sets using the relation above as its defining property, as follows.

The *abstract geodetic convexity* [20] $\mathcal{C}$ on X induced by M is the collection of geodetically-convex subsets of X, where a subset C of X is geodetically-convex provided $[x, y]_d \subseteq C$ for all x, y in C.

Using the definition above together with a specific distance d, one can, therefore, tell whether a set of points in the metric space endowed with the distance d is geodetically convex (with respect to that specific distance d). For example, in the Euclidean space, one can apply the definition of geodetic convexity with the Euclidean distance to see that the Euclidean ball is geodetically convex (with respect to the Euclidean distance, i.e., using the Euclidean segment).

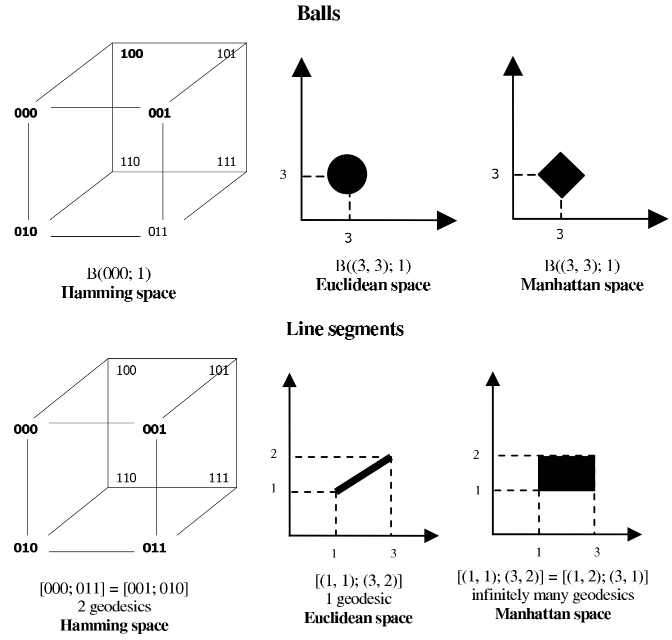

Figure 1: Examples of balls and segments

However, one can regard the notion of convex set from an abstract point of view by which a set of points in some *fixed but unspecified* metric space is geodetically convex with respect to the underlying (fixed but unspecified) distance associated with the metric space. An axiomatic approach to geodetically convex sets focuses on those properties of the collection of geodetically-convex sets that are independent on the specific distance considered and follow solely from the metric axioms and from the definition of abstract geodetic convexity *per se*. These properties are therefore valid for all metric spaces and associated space-specific geodetic convexities.

An alternative approach to generalizing the notion of convex set focuses on the property that the intersection of convex sets is a convex set, as follows. A family $\mathcal{X}$ of subsets of a set X is called *convexity on* X [20] if:

(C1) the empty set $\emptyset$ and the universal set X are in $\mathcal{X}$

(C2) if $\mathcal{D} \subseteq \mathcal{X}$ is non-empty, then $\bigcap \mathcal{D} \in \mathcal{X}$

(C3) $\mathcal{D} \subseteq \mathcal{X}$ is non-empty and totally ordered by inclusion, then $\bigcup \mathcal{D} \in \mathcal{X}$.

The pair $(X, \mathcal{X})$ is called *convex structure*. The members of $\mathcal{X}$ are called *convex sets*. By the axiom (C1) a subset A of X of the convex structure is included in at least one convex set, namely X. From axiom (C2), A is included in a smallest convex set, the *convex hull* of A: $co(A) = \bigcap\{C | A \subseteq C \in \mathcal{X}\}$. The convex hull of a finite set is called a *polytope*. The axiom (C3) requires *domain finiteness* of the convex hull operator: a set C is convex iff it includes $co(F)$ for each finite subset F of C. The following properties of the convex hull operator [20] will be useful:

(P1) $\forall A \subseteq X, co(co(A)) = co(A)$

(P2) $\forall A, B \subseteq X$, if $A \subseteq B$ then $co(A) \subseteq co(B)$

(P3) $\forall A, B \subseteq X$, if $A \subseteq co(B)$ then $co(A) \subseteq co(B)$

The two notions of convexity above are related as the collection $\mathcal{C}$ of geodetically convex sets under any metric d meet the convexity axioms. Notice, however, that if co denotes the convex hull operator of $\mathcal{C}$, then $\forall a, b \in X : [a, b]_d \subseteq co\{a, b\}$. So, there are metric spaces in which metric segments are not geodetically convex (i.e., in some spaces segments do not equal their convex hulls). In other words, an abstract metric segment is not necessarily geodetically convex. Also, there are metric spaces in which balls are not geodetically convex (e.g., in the Manhattan space, the Manhattan ball is not geodetically convex as the Manhattan segment between some pairs of points of the Manhattan ball is not completely included in the Manhattan ball). This exemplifies an important point: whereas the notion of abstract convexity appeals to the geometric intuition we all have for the Euclidean space, many familiar properties of the convexity in the Euclidean space do not hold across all metric spaces, hence do not hold for the abstract analogue of familiar shapes.

2.3 Euclidean and Hamming Convexities

When we specify the Euclidean distance in the definition of abstract geodesic convexity we obtain the traditional convexity for the Euclidean space [20]. However, the convexity for this space is not normally cast in terms of relations on distances between points (which we term *geometric characterization*) as the one obtained directly from the definition of geodesic convexity. Rather, it is expressed equivalently in algebraic terms using algebraic operations – sums of vectors and scalar products – which are well-defined on the Euclidean space but not on general metric spaces (which we term *algebraic characterization*). Naturally, this twofold characterization is made possible for the case of the Euclidean space as there is a natural one-to-one correspondence between points in space and real vectors (i.e., their cartesian coordinates). The algebraic characterization of convex sets and related notions for the Euclidean space is as follows. A point p is in the convex hull of a set of points S iff the coordinates of p can be obtained by a convex combination of the coordinates of the points in S. The segment between two points is the convex hull of its extremes (i.e., segments in the Euclidean space are convex).

Analogously to the Euclidean case, when we specify the Hamming distance on binary strings in the definition of abstract geodesic convexity we obtain the specific convexity for this space. Also in this case, we can characterize the convexity equivalently in algebraic terms using operations and notations which are well-defined on the underlying representation of points in space, which is, on binary strings. The algebraic characterization of convex sets and related notions for the Hamming space is as follows. Let $H(a, b)$ be the schema obtained from the binary strings a and b by position-wise inserting a '`*`' symbol where they mismatch and inserting the common bit otherwise (e.g., $H(0101, 1001) =$ `**01`). By abuse of notation, we consider a schema as being both a template and the set of strings matching the template. The binary string c is in the Hamming segment between the binary strings a and b iff c matches the schema $H(a, b)$ (e.g., 0001 is in the segment $[0101, 1001]$ as it matches the schema `**01` which can be verified using the definition of segment $d(0101, 0001)+d(0001, 1001) = d(0101, 1001)$). Every segment in the Hamming space is convex, because for $c, d \in H(a, b)$ the schema $H(c, d)$ can be obtained by changing some of the '`*`' symbols in $H(a, b)$ to 0 or 1, hence it is more specific than $H(a, b)$ (i.e., $H(c, d) \subseteq H(a, b)$). Every schema is a convex set as it corresponds to a segment between some pair of binary strings belonging to it. Every convex set is a schema because the set of all Hamming segments form the convexity structure on the Hamming space, as it is the product convexity of the trivial metric space [20]. Consequently, the intersection of two schemata is a schema or the empty set (e.g., `**101` $\cap$ `1**01` $=$ `1*101`) and the convex hull of a set of binary strings is the smallest schema (the schema matching the minimum number of strings) matching all of them (e.g., $co(0101, 1001, 0000) =$ `**0*`). In summary, in the Hamming space, the notions of segment, convex set and schema essentially coincide.

3. CONVEX EVOLUTIONARY SEARCH

3.1 Geometric operators

Geometric operators are search operators defined using geometric shapes to characterize the spatial relation between parents and offspring in the search space. Importantly, the shapes considered are defined in terms of distances between points in space. The geometric view of search operators formalizes and simplifies the relationship between representations, search operators and associated distance, and it equates their induced search space and fitness landscape to the traditional notions of neighbourhood space and fitness landscape.

Definition 1. *(Geometric crossover [10]) A recombination operator is a geometric crossover under the metric d if all offspring are in the d-metric segment between its parents.*

In a similar vein, geometric mutation is defined geometrically requiring that offspring are in a d-ball of a certain radius centered in the parent.

Notice that the definition is *representation-independent*, hence well-defined for any representation, as it depends on the underlying specific representation only indirectly via the metric d which is defined on the representation. This class of operators is really broad [3] [9]. For vectors of reals, various types of blend or line crossovers are geometric crossovers under Euclidean distance, and box recombinations and discrete recombinations are geometric crossovers under Manhattan distance. For binary and multary strings, all mask-based crossovers are geometric under Hamming distance. For permutations, PMX and Cycle crossover are geometric under swap distance and merge crossover is geometric under adjacent swap distance; other crossovers for permutations are also geometric. For genetic program trees, the family of homologous crossovers is geometric under structural Hamming distance. For biological sequences, various homologous recombinations that resemble more closely biological recombination at molecular level (as they align variable-length sequences on their contents, rather than position-wise, before

[3] The class of geometric crossover does not fully exhaust the range of crossover operators in common use. For example, sub-tree swap crossover for genetic program trees is provably not a geometric crossover under any metric.

swapping genetic material) are geometric under Levenshtein distance. Recombinations for several more complex representations are also geometric.

A more fine-grained definition of geometric crossover is possible by specifying a specific probability distribution of the offspring on the segment. For example, the uniform geometric crossover is defined as returning offspring sampled uniformly at random in the segment between parents. Uniform crossover on binary strings is known to be uniform geometric crossover for the Hamming distance [10]. The blend crossover on real vectors that samples offspring vectors uniformly at random in the line segment between parents is uniform geometric crossover for the Euclidean distance. Defining well-behaved probability distributions on general metric spaces needs a digression into measure theory and it is out of the scope of this framework. Pragmatically, uniform geometric crossover is well-defined on those metric spaces that admit a well-behaved notion of uniform distribution on segments, otherwise it is not definable. For most of the search spaces of interest uniform geometric crossover is well-defined.

A special class of probability distributions over the segment is that in which the probability of sampling an offspring z is a function of the distance $d(x, z)$ and $d(y, z)$ from its parents x and y. In this case, the points in the segment at the same distance from x are grouped into level sets according to the distance to x. This forms a partitioning on the points in the segment. Notice that any level set contains at least a point of the segment. So, if there are no points in the segment for a certain distance from the parent x, then there is no corresponding level set. Then the probability distribution specifies the probability of selecting a level set and then an offspring is sampled uniformly at random from the selected level set. For example, we could have a probability distribution that assigns the same probability of being selected to any level set i.e., one over the number of existing level sets in the segment. Note that this distribution equals the uniform geometric crossover in the Euclidean space (by appropriately considering limits and probability densities) as there is a single point at each distance level. However, it does not coincide to the uniform geometric crossover in the Hamming space as distance sets have different sizes (binomially distributed in the distance to the end-points of the segment).

Another special class of probability distributions over the segment is that of symmetric distribution probability in which the probability of obtaining the offspring z from the (ordered pair of) parents x and y is the same as when the role of the parents is reversed, i.e., $Pr(z|(x, y)) = Pr(z|(y, x))$. In practice, as the role of the parents as first or second parent is assigned at random with the same probability, the crossover operator can be always considered symmetric with probability distribution $f(z|x, y) = f(z|(x, y)) = f(z|(y, x)) = (Pr(z|(x, y)) + Pr(z|(y, x)))/2$.

3.2 Formal evolutionary algorithm and abstract evolutionary search

Geometric crossover and geometric mutation can be understood as functional forms taking the distance function d as a parameter. Therefore, we can see an evolutionary algorithm using these geometric operators as a function of the metric d too. That is, d can be considered as a parameter of the algorithm like any others, such as the mutation rate. However, notice the difference in the complexity of the objects passed as parameter: the mutation rate parameter takes values in the interval $[0, 1]$, that is, it is a simple real number, whereas the metric parameter takes values in the set of metrics, that is, it is a whole space.

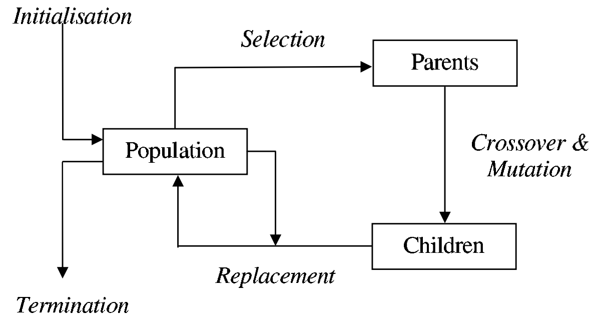

Figure 2: Evolutionary algorithm at a population level.

We can now look at an evolutionary algorithm as a function of the distance d from an abstract point of view. To do this, we do not consider any metric in particular and we treat an evolutionary algorithm using geometric operators as a formal specification of a representation/space independent algorithm with a well-defined formal semantic arising from the metric axioms only. The transition to this more general point of view is analogous to the transition from geodesic convexity with respect to a specific metric space to the notion of abstract geodesic convexity. We refer to an evolutionary algorithm seen according to the latter interpretation as a *formal evolutionary algorithm*. A different notion of formal search algorithm based on equivalence classes was introduced by Radcliffe and Surry [19].

Normally, an algorithm can actually be run only when all its parameters have been assigned a value. We call an algorithm with all its parameters specified, a fully-specified algorithm. However, a formal model of the algorithm can be used to infer the behavior of a partially specified algorithm in which some parameters are left unspecified. In other words, using a formal model one can "run" a partially specified algorithm and infer its abstract behavior, i.e., those behavioral properties common to all specific behaviors obtained by assigning all possible specific values to the parameter left unspecified. We term *abstract evolutionary search* the behavior of a formal evolutionary algorithm in which the underlying metric d is unspecified. As this behavior is inferred from the formal evolutionary algorithm and the metric axioms only, *it is the behavior of the formal evolutionary algorithm on all possible search spaces and associated representations.*

The abstract behavior of a formal evolutionary algorithm is an axiomatic object itself based on the metric axioms. In the following sections, we will show that the behavior of a formal evolutionary algorithm can be profitably described axiomatically using the language of abstract convexity.

3.3 Genetic operators at a population level

An evolutionary algorithm can be seen as repeating a loop of operations at the population level (see Figure 2). The cycle selection-crossover-mutation-replacement can be seen as the sequential functional application of these operators to a population returning another population.

Let S be the search space and N be the set of natural numbers. A population is a multi-set in which each candidate solution can have multiple occurrences. The following population operators are presented very generally and probably could virtually cover any conceivable variant of evolutionary algorithm structure. The fitness function f and distance function d, which are parameters of the population operators, are fixed but unknown functions:

selection $OP_{SEL} : N^S \rightarrow N^S$: the selection operator is a possibly stochastic operator that takes in input a population and returns a population in which some of the elements have been reduced or increased in frequency and other have been eliminated according to some criteria, possibly fitness-based.

crossover $OP_{XO} : N^S \times N^S \rightarrow N^S$: the crossover operator at a population level is an operator that takes in input a population and returns a population of offspring obtained by applying any geometric crossover operator under d (with any probability distribution) to pairs of elements in the input population any number of times.

mutation $OP_{MUT} : N^S \rightarrow N^S$: the mutation operator at a population level is an operator that takes in input a population and returns a population of offspring obtained by applying any geometric mutation operator under d (with any probability distribution) to any element in the input population any number of times. The mutation is non-degenerate when it has non-zero probability of producing offspring different from parents.

replacement $OP_{REP} : N^S \times N^S \rightarrow 2^N$: the replacement operator is the sequential application of a merge operation, which merges the two population in input (union of multi-sets) followed by a selection operation.

3.4 Abstract convex search theorem

DEFINITION 2. *(Convex operator): let S be the solution set, an operator $OP : 2^S \rightarrow 2^S$ that takes a subset $P \subseteq S$ as input and returns a subset $OP(P) \subseteq S$ is a convex operator iff $\forall P \subseteq S : OP(P) \subseteq co(P)$*

The notions of abstract convexity and convex operators naturally extend to multisets and stochastic operators. The definition of convex operator extends to multisets substituting multisets with their underlying sets. The definition of convex operator extends to stochastic operators by substituting $OP(P)$, that is a random variable, with its support set $Im(OP(P))$ that is the set of elements that have probability non-zero to be returned by $OP(P)$.

THEOREM 1. *The composition of convex operators is a convex operator.*

PROOF. Let OP and OP' be two convex population operators and $OP'' = OP' \circ OP$. To prove that the composition of two convex operators is a convex operator we need to prove that $\forall P : OP(P) \subseteq co(P) \wedge OP'(P) \subseteq co(P) \longrightarrow OP''(P) = OP'(OP(P)) \subseteq co(P)$. By definition of convex population operator, it follows $OP(P) \subseteq co(P)$ and $OP'(OP(P)) \subseteq co(OP(P))$. From the property of convex hull $OP(P) \subseteq co(P)$ implies $co(OP(P)) \subseteq co(P)$. Hence, $OP'(OP(P)) \subseteq co(P)$. □

THEOREM 2. *(Convexity of genetic operators at a population level) Selection, Crossover (as in Section 3.3) and Replacement (with the offspring population in the convex hull of the parent population) are convex population operators. Non-degenerate Mutation is not a convex operator.*

PROOF. *Selection*: let $P' = OP_{SEL}(P)$. As $P' \subseteq P$ by selection and $P \subseteq co(P)$ by a property of the convex hull then $OP_{SEL}(P) \subseteq co(P)$. Selection is a convex operator.
Crossover: let $C = OP_{XO}(P)$. Every offspring in C is in the segment between two parents in P. For the geodesic convexity, for any $x, y \in P$ we have $[x, y] \subseteq co\{x, y\} \subseteq co(P)$, hence $OP_{XO}(P) = C \subseteq co(P)$. The crossover operator is a convex operator.
Replacement: let $P' = OP_{REP}(P, C)$. We say that OP_{REP} is convex if when $C \subseteq co(P)$ then $OP_{REP}(P, C) \subseteq co(P)$. Since $C \subseteq co(P)$ then $co(P \cup C) \subseteq co(P \cup co(P)) = co(co(P)) = co(P)$, hence $P' = OP_{REP}(P, C) \subseteq co(P)$. The replacement operator is a convex operator.
Mutation: every convex operator returns points within the convex hull of the input set. The convex hull of a single point is the single point itself. So, when the input set includes a single point, the output set of any convex operator must be the point itself. Mutation applied to a single point p may produce points different from p, hence it is not a convex operator. □

THEOREM 3. *(Abstract convex evolutionary search) Let P_n be the population at time n. For any evolutionary algorithm repeating the cycle selection, crossover, replacement we have $co(P_{n+1}) \subseteq co(P_n) \subseteq \cdots \subseteq co(P_1) \subseteq co(P_0)$*

PROOF. The compound operator
$OP = OP_{REP} \circ (OP_{ID}, OP_{XO} \circ OP_{SEL})$ that is equivalent to the sequential application of selection, crossover and replacement is a convex operator (OP_{ID} is the identity operator that outputs its own input). This is because OP_{SEL} and OP_{XO} are convex operators hence $OP_{XO} \circ OP_{SEL}$ is a convex operator for the composition of convex operators theorem. Hence OP_{REP} is also a convex operator because its second argument $OP_{XO} \circ OP_{SEL}$ is in the convex hull of its first argument OP_{ID} . Hence, for the composition of convex operators theorem, OP is a convex operator. Since $P_{n+1} = OP(P_n)$ then $P_{n+1} \subseteq co(P_n)$ and consequently $co(P_{n+1}) \subseteq co(P_n)$. Then the chain of nested inclusions is true by induction. □

Theorem 3 is very general. *An evolutionary algorithm using geometric crossover with any probability distribution, any representation, any problem, any selection and replacement mechanism, does the same form of convex search.* Population size can vary over time and evolutionary search is still convex.

3.5 Convex evolutionary search in Euclidean and Hamming spaces

Theorem 3 applies to all metric spaces. It gives an abstract geometric description of the search that does not depend on any specific distance. In the following, we visualize the abstract convex search for the specific case of the 2-dimensional Euclidean space. This leads to a very simple and useful description of it which illustrates geometrically why the theorem 3 holds. Indeed, the geometric reason this theorem holds for the 2-dimensional Euclidean case is the same as the reason it holds in general metric spaces.

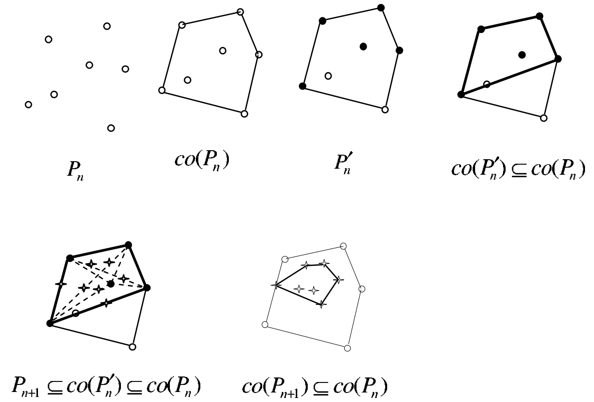

Figure 3: Convex Evolutionary Search.

Figure 3 illustrates the convex search. The hollow circles represent individuals in the population at time n, solid circles represent selected individuals (parents), while crosses represent individuals in the populations at time $n+1$. The thin solid lines connecting hollow circles represent the boundaries of the convex hull formed by individuals in the population at time n. The thick solid lines connecting solid circles represent the boundaries of the convex hull formed by selected individuals from the population at time n. The solid lines connecting crosses represent the boundaries of the convex hull formed by individuals in the population at time $n+1$. The broken lines connecting solid circles include all possible offspring of the selected individuals from the population at time n by applying geometric crossover (offspring are in the segment).

An evolutionary algorithm with mutation does not do convex search. It is the non-convexity of mutation that can cause the evolutionary search not to be convex as this allows for the possibility that an offspring can be produced outside the convex hull of the previous population.

The convex evolutionary search theorem shows that the convex hulls of the populations at successive iterations form a nested chain of inclusions. So, the search never diverges. Since the inclusions in the nested chain are not strict, the search either (i) converges to a fixed-point, or (ii) stops its convergence to a single set of points, e.g., for deterministic degenerate forms of selection, crossover and replacement that do not change the current population; or (iii) stops its convergence to a (possibly infinite) orbit of sets of points that have the same convex hull, e.g., when the individuals at the corners of the convex hull of the current population are always passed on to the next population, while the other individuals in the population are changed.

It is interesting to look at an example of the instantiation of the abstract convex search to the case of Hamming space on binary strings. Set $P_n = \{00010, 01100, 01110, 10000\}$. Then $co(P_n) =$ `****0`. Selection is applied and say that only the last string is discarded, so the set of parents is $P'_n = \{00010, 01100, 01110\}$. Then $co(P'_n) =$ `0***0` which gives `0***0` $\subseteq$ `****0`. Recombination is then applied using the geometric crossover specified to the Hamming space (e.g., uniform crossover) and say we have the following recombinations: $CX(00010, 01100) \rightarrow 01110, 01000$, $CX(00010, 01110) \rightarrow 01010$ and $CX(01100, 01110) \rightarrow 01100$. Let us say we have a generational replacement scheme so that the population of offspring replaces the population of parents, so $P_{n+1} = \{01110, 01000, 01010, 01100\}$.
Then $P_{n+1} \subseteq co(P'_n)$ as all offspring in P_{n+1} match the schema `0***0`. Then $co(P_{n+1}) =$ `01**0` which gives $co(P_{n+1}) \subseteq co(P_n)$ as the schema `01**0` is more specific than the schema `****0`.

4. CONVEX SEARCH AND CONCAVE FITNESS LANDSCAPE

4.1 Matching abstract search and abstract fitness landscape

The NFL theorem [21] implies that a search algorithm must be well-matched with a certain class of fitness landscapes respecting some conditions to perform on average better than random search. As a consequence, any non-futile theory which aims at proving performance better than random search of a class of search algorithms needs to indicate with respect to what class of fitness landscapes. Therefore, an important question is: are there general conditions on the fitness landscape that guarantee good performances of the convex search for any space/representation?

Since, at an abstract level, all evolutionary algorithms encompassed by the abstract evolutionary search theorem presents a unique behavior, it is reasonable to put forward the hypothesis that there should exist a general class of fitness landscapes that is well-matched to the evolutionary search as a whole. Otherwise stated, as evolutionary algorithms at heart present a common behavior, they should produce good performance on the same type of fitness landscape *for essentially the same underlying reason*. To make sense, the level of abstraction of the condition on the fitness landscape characterizing this class of fitness landscapes, whatever this class could be, must match the level of abstraction of the evolutionary search. So, the definition of the condition must be based on the distance of the search space, but, like the abstract convex search, it must be meaningful *per se* without referring to any specific distance. In other words, it must be a condition that matches convex search and fitness landscape at an abstract level. Figure 4 illustrates the envisioned functional relationship between search algorithm (SA), fitness landscape (FL), search behavior (SB) and search performance (SP) and their abstract counter-parts, formal search algorithm (FSA), abstract fitness landscape (AFL), abstract search behavior (ABS) and abstract search performance (ASP). The horizontal arrows in the bottom means that the algorithm SA is fed with the parameter fitness landscape FL which when run together give rise to the search behavior SB that produces the search performance SP. The horizontal arrows in the top mirror those in the bottom and depict analogous relations at an abstract level in which the underlying distance d is left unspecified. The vertical arrows relate abstract and concrete levels by functional application of the functional forms in the top with a specific distance d.

In the following sections, we will start investigating how to define classes of abstract fitness landscapes and how they affect abstract performance when matched with the abstract convex search. Abstract performance can be constant with the parameter d and hold across all metric spaces unchanged, or can be a general expression in which specific character-

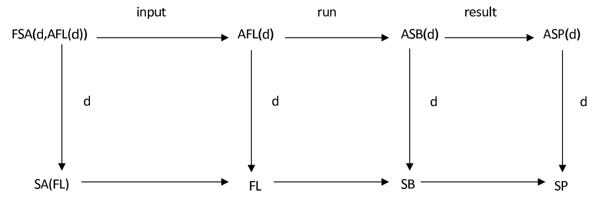

Figure 4: Functional relations between search algorithm, fitness landscape, search behavior, search performance and their abstract counter-parts.

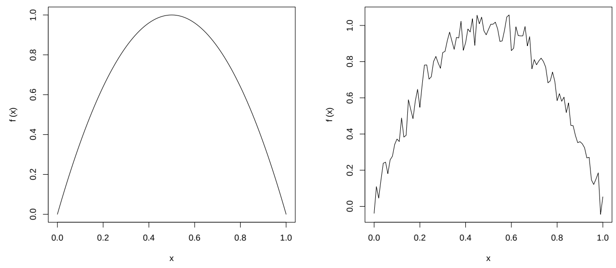

Figure 5: Examples of concave function (left) and function with a concave trend (right).

istics of the underlying metric d (e.g., dimensionality) are used as parameters.

4.2 Concave fitness landscapes

In this and in the following sections, we will explore several notions of concave fitness landscape progressively improving the scope of application.

The convex search of an evolutionary algorithm suggests a class of functions, which *suitably generalized*, could give rise to a class of abstract fitness landscapes well-matched with the abstract convex search for which we may obtain good performance: concave functions and approximately concave functions[4] (see Figure 5. In this section, we consider concave functions and in section 4.4 we will consider approximately convex functions). Note, however, that this intuition arises from the notion of convexity in the Euclidean space, and may easily not hold true across all metric spaces i.e., may not hold at an abstract level. Furthermore, even when the generalization to general metric spaces is possible, a great deal of caution is required, as a generalized notion of concave function which is meaningful when specified to continuous spaces may lead to only degenerate cases of little interest, when specified to combinatorial spaces. In this section, we will start considering candidate generalizations of the notion of concave functions to general metric spaces and show how the (abstract) performance of (abstract) convex search is bounded on these classes of (abstract) fitness landscapes. In the introduction, we have mentioned that we are interested in generalizing the "concave trend" character of a concave function, rather than its "unimodal" character. This is because it is known experimentally that, assuming minimization, global convexity of the fitness landscape [2] (i.e., a globally convex trend) is beneficial for the performance of evolutionary algorithms (also known as the "big valley" hypothesis) and many important combinatorial problems have been shown to be globally convex [7]. In the next section, we will extend the generalized notion of concave functions to generalized notion of functions with a concave trend in the attempt to characterize formally the notion of global concavity (and convexity).

For the following definitions see e.g. [3]. A real-valued function f defined on any convex subset C of some vector space is called concave iff the Jensen's inequality holds for any two points $x, y \in C$, i.e., for any $t \in [0, 1]$, $f(tx + (1 - t)y) \geq tf(x) + (1-t)f(y)$, which is, the chord connecting any two points of a concave function lies below or on the graph of the function. A function f is convex iff $-f$ is concave. If a function is convex and concave at the same time it is an affine function (e.g., any linear function in some vector space is affine).

[4]We assume maximization. For minimization problems, the classes of functions to consider are convex functions and approximately convex functions.

There are a number of ways to generalize convex and concave functions to more general spaces than vector spaces to discrete spaces and to general metric spaces (see e.g., the monograph [16] on this topic with focus on generalizing results of convex optimization). We will consider generalizations which can be used as basis for the generalization of convex and concave trends. A concave function can be generalized to a metric space [17] (i.e., a concave fitness landscape) as follows. Let C be a geodetically convex set on a metric space endowed with distance d, then $f : C \to R$ is concave if for any two points $x, y \in C, d(x, y) \neq 0$ and bounded, and for any $z \in [x, y]_d$, $f(z) \geq \frac{d(y,z)}{d(x,y)} f(x) + \frac{d(x,z)}{d(x,y)} f(y)$. Analogously, the function is convex if the inequality obtained by changing $\geq$ to $\leq$ holds. This is a proper generalization of concave function, as when the space considered is a vector space endowed with the Euclidean distance, the generalized notion of concavity reduces to the classical notion [17]. Later in this section, we will illustrate with an example how this definition applies to combinatorial spaces. However, first we present a result that makes this definition potentially interesting for our framework.

We first need a condition of regularity of the underlying space. We say that a metric space (S, d) has symmetric segments if for any $x, y \in S$ the sizes of the distance level partitions on the segment from x i.e., the collection of sets $[x, y]_t = \{z : z \in [x, y] \wedge d(x, z) = t\}$ indexed by t (the admissible values of t are those which do not return an empty partition), correspond to the sizes of the distance level partitions from y, i.e., for all admissible t $[x, y]_t = [y, x]_t$.

THEOREM 4. *On a concave fitness landscape f endowed with any distance d with symmetric segments, the expected fitness of the offspring z obtained by recombining any two parents x and y using the geometric crossover associated with the distance d with any symmetric offspring probability distribution is not less than the average fitness of its parents, i.e.,* $E[f(z)|x, y] \geq \frac{f(x)+f(y)}{2}$.

PROOF. We prove the theorem for the case of discrete spaces. The proof for continuous spaces is analogous and can be obtained by replacing probability mass functions with probability density functions and summations with integrals. A geometric crossover, given parents x and y, returns an offspring $z \in [x, y]$ according to a certain probability distribution over the segment $Pr(z|d(x, z) = t)$ of sampling a point z at a given distance t from x (first argument of the crossover). Hence, given the parents x and y the expected fitness of the offspring is $E[f(z)|x,y] \geq \sum_{t=0\ldots d(x,y)} Pr(z|d(x,z)=t) \cdot ((1-\frac{t}{d(x,y)})f(x)+\frac{t}{d(x,y)}f(y))$. Exchanging the parents (x as second argument of the crossover and y as first argument) the probability distribution over the segment has origin in y and the expected offspring of the offspring becomes

$$\begin{aligned}
E[f(z)|x,y]' &\geq \sum_{t=0\ldots d(x,y)} Pr(z|d(x,z)=t)' \cdot \\
&\quad ((1-\frac{t}{d(x,y)})f(x)+\frac{t}{d(x,y)}f(y)) \\
&= \sum_{t=0\ldots d(x,y)} Pr(z|d(x,z)=d(x,y)-t) \cdot \\
&\quad ((1-\frac{t}{d(x,y)})f(x)+\frac{t}{d(x,y)}f(y)) \\
&= \sum_{t'=0\ldots d(x,y)} Pr(z|d(x,z)=t') \cdot \\
&\quad ((1-\frac{d(x,y)-t'}{d(x,y)})f(x)+\frac{d(x,y)-t'}{d(x,y)}f(y)) \\
&= \sum_{t'=0\ldots d(x,y)} Pr(z|d(x,z)=t') \cdot \\
&\quad (\frac{t'}{d(x,y)}f(x)+(1-\frac{t'}{d(x,y)})f(y)).
\end{aligned}$$

$E[f(z)|x,y]+E[f(z)|x,y]' \geq \sum_{t=0\ldots d(x,y)} Pr(z|d(x,z)=t) \cdot$
$((1-\frac{t}{d(x,y)}+\frac{t}{d(x,y)})f(x)+(\frac{t}{d(x,y)}+1-\frac{t}{d(x,y)})f(y))$
$=\sum_{t=0\ldots d(x,y)} Pr(z|d(x,z)=t)\cdot(f(x)+f(y))$
$=f(x)+f(y)$.
For symmetry of the probability distribution
$E[f(z)|x,y]'=E[f(z)|x,y]$ then $E[f(z)|x,y] \geq \frac{f(x)+f(y)}{2}$. □

When the symmetry condition on the segment does not hold, the average of the fitness of the parents may be skewed toward one or the other parent. However, we always have $E[f(z)|x,y] \geq \min\{f(x),f(y)\}$. The theorem above can be easily extended to populations.

COROLLARY 1. *On a concave landscape, by applying geometric crossover on pairs of parents sampled uniformly at random (with replacement) from any population of parents, the expected average fitness of the offspring population is not less than the average fitness of the parent population.*

PROOF. The expected average fitness of the offspring population obtained by sampling at random the parent population Pop of size n is

$$\begin{aligned}
E[f(z)] &= \sum_{x,y\in Pop} \frac{1}{n^2} E[f(z)|x,y] \\
&\geq \frac{1}{2n^2} \sum_{x,y\in Pop} (f(x)+f(y)) \\
&= \frac{1}{2n^2} (\sum_{x,y\in Pop} f(x) + \sum_{x,y\in Pop} f(y)) \\
&= \frac{1}{2n^2} (2n \sum_{x\in Pop} f(x)) \\
&= \frac{\sum_{x\in Pop} f(x)}{n}.
\end{aligned}$$

□

It is noticeable that the theorem above is a statement about the one-step performance of the formal evolutionary algorithm that holds in a very general setting. Notice also that the formal evolutionary algorithm on concave landscapes, on average, makes steady progress or, in the worst-case, does not get worse, *even in lack of selection*[5] at any stage of the search process, i.e., $\forall n : E[f(P_{n+1})] \geq E[f(P_n)]$. This is a rather strong statement about the one-step performance because, as a norm, the mean fitness of the offspring is less than the average fitness of their parents, and selection (or selective replacement) is a necessary ingredient to obtain progress [6].

We now illustrate with an example how the definition of convex function introduced earlier applies to combinatorial spaces by showing how it can be used to check whether a fitness landscape with any underlying metric d is convex, concave, affine or none of them. Let us consider the fitness landscape obtained by the One-Max function f on the space of binary strings endowed with the Hamming distance. Let $x = 000111$ and $y = 001100$. So we have $f(x) = 3$ and $f(y) = 2$ and $d(x,y) = 3$. The points in the segment $[x,y]$ are those matching the schema $H(x,y) =$ `00*1**`. Then, if f is concave, from the definition of generalized concavity, for each string z matching the schema we must have $f(z) \geq \frac{d(y,z)}{d(x,y)}f(x) + \frac{d(x,z)}{d(x,y)}f(y)$. Let us consider the point in the segment $z = 000100$ we have $f(z) = 1$, $d(x,z) = 2$ and $d(y,z) = 1$. By substituting these values in the inequality defining concave function, we obtain that it does not hold true. So, One-Max is not a concave function in the above sense. It is also possible to show that One-Max is not a convex function either as for the point in the segment $z = 001111$ the defining inequality for convex function does not hold. Unfortunately, the above scenario recurs over and over. It turns out that the only fitness landscapes based on binary strings endowed with the Hamming distance which are concave (and convex) in the above sense are constant landscapes. So, this makes the above definition of concavity unsuitable for our framework as it does not encompass interesting functions on the Hamming space.

So, are there alternative definitions of generalized concave function which encompass interesting fitness landscapes on the Hamming space? One possibility is to relax suitably the definition of concave function retaining the results of theorem 4 and its corollary. Interestingly, the theorem 4 holds when the average fitness of each distance level partition of the segment – rather than the fitness of any point in each distance level partition – is greater than the corresponding

[5]When parents are selected independently and with the same distribution increasing the selection pressure can lead to an increase of the average population fitness. In case of adversary selection schemes, in which worse individuals are preferred to better individuals, the one-step performance can decrease due to selection. Naturally, adversary selection is never used.

[6]On the general class of concave landscapes, this is the strongest lower bound one can obtain. Clearly, such a result does not lead to interesting lower-bounds for the n-step performance. However, one could consider restricted classes of concave landscapes with given curvature (e.g., concave landscapes which are also bi-Lipschitz with constant k), and derive stronger bounds for the one-step performance as a function of the degree of curvature that could lead to interesting bounds for the n-step performance. We will consider this possibility in future work.

linear combination of the fitness of the endpoints for that distance level. Therefore this weaker condition can be used to characterize a larger class of generalized concave functions for which theorem 4 and its corollary hold. In this weaker sense, the One-Max landscape turns out to be affine. So, the theorem 4 and its corollary apply to it with an equality sign rather than a greater and equal. In the next section, we consider a weaker form of concave functions – quasi-concave functions – which can be generalized to general metric spaces more naturally than concave functions.

4.3 Quasi-concave fitness landscapes

For the following definitions see e.g. [3]. A real-valued function f defined on any convex subset C of some vector space is called quasi-concave iff for any real number a the inverse image of the set of the form $(a, +\infty)$ is a convex set (notice that this implies that sets corresponding to larger values of a are included in the sets corresponding to smaller values of a), and quasi-convex iff the inverse image of any set of the form $(-\infty, a)$ is a convex set. A function that is both quasi-convex and quasi-concave is quasi-linear. All convex functions are also quasi-convex, but not all quasi-convex functions are convex, so quasi-convexity is a weak form of convexity. An analogous relation between concave and quasi-concave functions hold. Quasi-concave and quasi-convex functions can be equivalently characterized using inequalities. A real-valued function f defined on any convex subset C of some vector space is quasi-concave iff for any two points $x, y \in C$ for any $t \in [0, 1]$, $f(tx+(1-t)y) \geq \min(f(x), f(y))$, and quasi-convex iff for any two points $x, y \in C$ for any $t \in [0, 1]$, $f(tx + (1 - t)y) \leq \max(f(x), f(y))$.

We can readily generalize quasi-concave functions to metric spaces by requiring that the inverse image of any set of the form $(a, +\infty)$ is a *geodetically* convex set. As geodetically convex sets for the Euclidean distance reduce to traditional convex sets, the one above is a proper generalization of quasi-concave function. As the level sets are geodetically convex and form a nested chain of inclusions for increasing values of a, the function value at any point of a segment cannot be lower than the minimum of the function value of its endpoints. So, analogously to the traditional case we have that for generalized quasi-concave functions hold: $f(z) \geq \min(f(x), f(y))$ for any $z \in [x, y]$. Quasi-convex and quasi-linear functions can be generalized to metric spaces analogously.

As for the case of the generalization of concave functions, also for the generalization of quasi-concave functions we can bound the expected fitness of the offspring population on the average fitness of the parent population. From the definition of quasi-concave landscape, it is clear that the fitness of any offspring cannot be less than the the minimum fitness in the parent population. However, it is possible to have a stronger result considering order statistics, which involve the distribution of the min, second-to-last, ..., median, ... max in a fixed set of samples. The line of reasoning is as follows. As parents that undergo geometric crossover can be thought as of being selected independently from the mating-pool, the fitnesses of the two parents are i.i.d. random variables with distribution equalling the fitness distribution of the mating-pool. The expected fitness of the offspring is then larger than the expected fitness of the first order statistics of the two random variables. To form an understanding of how the lower bound of the expected fitness offspring population is less than the average fitness of the parent population, we can linearly rank the parents in the mating-pool in the interval $[0, 1]$ so that rank 1 is the best, rank 0 is the worst, and rank 0.5 is the median. Then the expected rank of the offspring population is lower-bounded by the expected value of the first order statistic of the uniform distribution on $[0, 1]$, which is $1/3$. An analogous result holds for the generalization of quasi-convex function. Notice also that on quasi-linear functions the maximum fitness of the offspring population is upper-bounded by the maximum fitness of the parent population, so in this case, evolution will not take place because there is no chance for offspring to improve over the parents. So, assuming maximization, for improvement to be possible we need to have a quasi-concave landscape which is not quasi-linear.

Quasi-concave landscapes are interesting when considered together with combinatorial spaces because they give rise to discrete sets of fitness values, in other words, fitness functions on combinatorial spaces are quantized functions. This property allows us to check quasi-concavity of a landscape by checking directly one by one the convexity of all level sets. Also, we can use this property to easily construct quasi-concave functions for any representation and metric space once we have determined the specific form of geodetically convex sets. In the following, we illustrate this for the case of the Hamming space on binary strings. The idea for the construction procedure of a quasi-convex landscape is to build directly a nested chain of convex level sets starting from the largest level set with the minimum fitness value which is then recursively partitioned in two parts: (i) a convex set containing the next fitness level or fitness levels with higher values, and (ii) the complement of that set with respect to the convex set at the current level containing points whose fitness values are matching the current fitness level. Let us assign fitness equal or greater than zero to all points in space. So, for any x_0 matching the schema `******`, which is the current level set, we have $f(x_0) >= 0$. Let us consider the convex subset `0*****` of `******` and let us assign to any x_1 matching this schema fitness values of one or greater, i.e., $f(x_1) >= 1$. The complement set of `0*****` with respect to `******` is `1*****` for which all matching strings keep the fitness of the current level set, which is zero. So, the quasi-concave function we will obtain has fitness zero for any string starting with 1 and fitness strictly larger than 1 for any string starting with 0. Let us now consider `0*****` which is the current level set which we need to partition. We chose the convex subset `01****` of `0*****` to have fitness values equal or larger than 1, and its complement set in `0*****`, which is `00****`, to have fitness values at the current level, which is 1. So, the quasi-convex function will return fitness 1 for all strings starting with 00. Then the procedure continues so forth and so on partitioning the current sets until a convex set is reached which contains a single element, or at any time if one decides to assign the current fitness level also to the chosen convex set, so obtaining a landscape with a plateau. For example, continuing the procedure above we could construct the following function: $f(\texttt{1*****}) = 0, f(\texttt{00****}) = 1, f(\texttt{011***}) = 2, f(\texttt{0101**}) = 3, f(\texttt{01000*}) = 4, f(\texttt{010010}) = 5, f(\texttt{010011}) = 6$. As the Leading-One landscape can be built in this way, it is a quasi-concave landscape. It is possible to show that One-Max is not a quasi-concave landscape because its level sets are Hamming balls, which unlike Euclidean balls, are not geodetically convex.

Analogously to traditional quasi-concave functions, also for quasi-concave landscapes a number of combinations and transformations on quasi-concave landscapes produce quasi-concave landscapes. For example, if f, g are quasi-convex defined on the same domain, the following landscapes are quasi-convex: $\alpha f + \beta$ with α, β real numbers, $\min[f, g]$, and $h(f)$ with h monotonic increasing.

4.4 Fitness landscapes with concave trend

As we mentioned in the introduction, there is a well-known notion of global convexity of the fitness landscape [2] that is known experimentally to be beneficial for the performance of evolutionary algorithms. Interestingly, fitness landscapes normally associated with many important combinatorial problems have been shown to be globally convex [7]. Therefore, a theory which characterizes the performance of evolutionary search on this class of fitness landscapes would be very relevant to practice. In this section, we will start characterizing the notion of globally convex landscape formally and in a general way, by suitably extending the notions of convex and quasi-convex landscapes presented earlier. Naturally, the practical utility of the formalization proposed is bound to the extent to which fitness landscapes arising in practice fit nicely these formal notions. In future work, we will investigate this issue thoroughly and, if necessary, propose a refined definition of concave trend which suits better these landscapes.

In the literature, there are a number of ways of weakening the concavity/convexity requirement of a function to approximate concavity/convexity. A simple approach to defining approximately concave functions [5] (see Figure 5 (right)) is as follows. Let $\epsilon \geq 0$. A real-valued function f defined on any convex subset C of some vector space is called ϵ-concave if for any two points $x, y \in C$, for any $t \in [0, 1]$, $f(tx+(1-t)y) \geq tf(x)+(1-t)f(y)-\epsilon$. This is a well-studied and interesting class of functions whose concavity approximation is controlled by the parameter ϵ, the smaller the better the approximation. Interestingly, a function $f = g + h$ obtained from a concave function g and a bounded perturbing function h such that $\forall x : |h(x)| \leq \epsilon$ is 2ϵ-concave [5].

A ϵ-concave function can be generalized to a metric space analogously to the case of concave functions, as follows. Let C be a convex set on a metric space endowed with distance d, then $f : C \to R$ is ϵ-concave if for any two points $x, y \in C, x \neq y$ and for any $z \in [x, y]$, $f(z) \geq \frac{d(y,z)}{d(x,y)} f(x) + \frac{d(x,z)}{d(x,y)} f(y) - \epsilon$. As for the Euclidean case, an analogous result on the sum of a concave landscape with a bounded perturbing landscape is a 2ϵ approximately concave landscape. Also, the following result which extends theorem 4 to ϵ-concave landscape holds (as the constant ϵ can be pulled out from all summations).

THEOREM 5. *On a ϵ-concave fitness landscape, the expected fitness of the offspring $E[f(z)]$ obtained by recombining randomly selected parents using geometric crossover from a population with average fitness $\bar{f}$ is $E[f(z)] \geq \bar{f} - \epsilon$.*

It is noticeable that the parameter controlling the concavity of the landscape has a straightforward impact on the one-step performance and, inductively, on the overall performance of the formal evolutionary algorithm: the less the concavity approximation, the lower the performance can become.

Whereas the above class of approximately concave landscapes may not be interesting for combinatorial spaces, as it is based on a class which is not interesting on the Hamming space, weaker versions on this class may encompass interesting classes of fitness landscapes. For example, theorem 5 may well hold for the average version of concave landscapes, so that the theorem would give a lower-bound on the fitness of the offspring for perturbed One-Max landscapes.

We can also define a class of approximately quasi-concave fitness landscapes, as follows. Let C be a convex set on a metric space endowed with distance d, then $f : C \to R$ is ϵ-quasi-concave if for any two points $x, y \in C : f(z) \geq \min(f(x), f(y)) - \epsilon$ for any $z \in [x, y]$. The level sets of this class of functions are not convex, however they form a nested chain of inclusions, as level sets do so for any function. For $a \leq b$, denoting the corresponding level sets $ls(a)$ and $ls(b)$, in general we have $ls(a) \supseteq ls(b)$ but the relation $ls(a) \supseteq co(ls(b))$ does not necessarily hold in general, but it always holds for quasi-concave landscapes. However, this holds also for ϵ-quasi-concave landscapes when $b - a \geq \epsilon$. Analogously to the case of quasi-concave landscapes, this characterization in terms of level sets can be used to construct ϵ-quasi-concave landscapes based on combinatorial spaces or to check for what ϵ a landscape is ϵ-quasi-concave. Also, notice that a result relating the expected fitness of the offspring population with the fitness of the parent population as a function of ϵ holds on this class of landscapes.

Interestingly, the classes of ϵ-concave landscapes and ϵ-quasi-concave landscapes are flexible classes as any fitness landscape is ϵ-concave and ϵ-quasi-concave for ϵ large enough. The practical relevance of these classes is therefore bound to the extent to which fitness landscapes arising in practice fit these classes for small values of ϵ.

Another interesting consequence of the flexibility of these classes of fitness landscapes is that to some extent any fitness landscape is both approximately concave and approximately convex, with in general different degrees for concavity and convexity. Since for the case of convexity one can determine an upper-bound of the expected fitness of the offspring population with respect to of the average fitness of the parent population, from the knowledge of the degrees of convexity and concavity of a landscape, one can obtain a range for the expected fitness of the offspring population. This may allow us to characterize the performance for n-steps rather than for a single-step of the evolution, perhaps in the average case.

Finally, it would be interesting to consider approximated concave landscapes with ϵ being a random variable rather than a constant. This would define a statistical model of fitness landscapes which may be more suitable to study average-case performance rather than worst-case.

5. CONCLUSIONS

It is not at all clear that a unified theory of evolutionary algorithms across representations is possible. This is an important open research question. The main contribution of this paper is to start shaping the foundational concepts of a possible approach towards such a theory.

We have started developing the theory by making use of mathematical abstraction in the form of an axiomatic geometric language, based on the notion of geometric search operators, that encompasses all representations. We have defined the notion of abstract search behavior and used it to prove that all evolutionary algorithms with geometric crossover across representations do a form of abstract convex search. This is a significant result as it elicits a common behavioral identity of all evolutionary algorithms (without mutation) across representations.

The unity in behavior of evolutionary algorithms calls for the existence of a unique abstract underlying condition on the fitness landscape for which all evolutionary algorithms perform well. Convex search naturally suggests that the concavity of the fitness landscape may be the key condition for obtaining good performance of evolutionary algorithms. We have generalized to general metric spaces a number of notions of concave functions and considered their suitability for the framework. We have shown that quasi-concave functions may be a natural choice for fitness landscapes based on combinatorial spaces. However, further study is required to confirm the suitability of this notion.

We have proved a general result showing a direct relationship between the degree of concavity of a fitness landscape and its impact on the one-step performance of evolutionary algorithms. This may be of direct relevance to combinatorial problems as many are known to be associated with fitness landscapes which are globally convex. However, it remains to be seen if the formalized notion of concavity aligns well with the empirical notion of global convexity. Investigating this and refining the formal definitions is an important piece of future work.

At the moment is still unclear how far the theory can be brought forward at this level of abstraction. In future work, we will investigate to what extent results about convergence to the optimum, convergence rate and run-time analysis can be obtained at this level of abstraction. Finally, as an important piece of future work, we will study the effect of mutation on convex search and how to accommodate it within this general framework.

6. REFERENCES

[1] Blumental and Menger. *Studies in geometry*. Freeman and Company, 1970.

[2] K. D. Boese, A. B. Kahng, and S. Muddu. A new adaptive multi-start technique for combinatorial global optimizations. *Operations Research Letters*, 15:101–113, 1994.

[3] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[4] K. DeJong. *Evolutionary computation: a unified approach*. The MIT Press, 2006.

[5] J. W. Green. Approximately convex functions. *Duke Mathematical Journal*, 19:499–504, 1952.

[6] W. Langdon and R. Poli. *Foundations of Genetic Programming*. Springer-Verlag, 2002.

[7] P. Merz and B. Freisleben. Fitness landscapes and memetic algorithms. In D. Corne, M. Dorigo, and F. Glover, editors, *New ideas in optimization*. McGraw-Hill, 1999.

[8] E. H. Moore. Definition of limit in general integral analysis. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 1, pages 628–632, 1915.

[9] A. Moraglio. *Towards a geometric unification of evolutionary algorithms*. PhD thesis, University of Essex, 2007.

[10] A. Moraglio and R. Poli. Topological interpretation of crossover. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1377–1388, 2004.

[11] K. Murota. Discrete convex analysis. *Mathematical Programming*, 83(1-3):313–371, 1998.

[12] N. Radcliffe. Equivalence class analysis of genetic algorithms. *Complex Systems*, 5:183–205, 1991.

[13] C. M. Reidys and P. F. Stadler. Combinatorial landscapes. *SIAM Review*, 44:3–54, 2002.

[14] F. Rothlauf. *Representations for Genetic and Evolutionary Algorithms*. Springer, 2002.

[15] J. E. Rowe, M. D. Vose, and A. H. Wright. Group properties of crossover and mutation. *Evolutionary Computation Journal*, 10(2):151–184, 2002.

[16] I. Singer. *Abstract Convex Analysis*. Wiley, 1997.

[17] V. P. Soltan. *Introduction to the Axiomatic Theory of Convexity*. Shtinitsa, 1984.

[18] C. R. Stephens and A. Zamora. EC theory: A unified viewpoint. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1394–1405, 2003.

[19] P. D. Surry and N. J. Radcliffe. Formal algorithms + formal representations = search strategies. In *Proceedings of Parallel Problem Solving from Nature Conference*, pages 366–375, 1996.

[20] M. van de Vel. *Theory of Convex Structures*. Elsevier, Amsterdam, 1993.

[21] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transaction on Evolutionary Computation*, 1(1):67–82, 1996.