**GENETIC ALGORITHMS**

1. **Genetic algorithms (GAs): basic form**

A generic GA (aka evolutionary algorithm [EA]) assumes a discrete search space *H* and a function

 ,

where *H* is a subset of the Euclidean space.

The general problem is to find

 ,

where *X* is a vector of the decision variables and *f* is the objective function.

With GAs it is customary to distinguish *genotype*–the encoded representation of the variables–from *phenotype*–the set of variables themselves. The vector *X* is represented by a string (or *chromosome*) *s* of length *l* made up of symbols drawn from an alphabet *A* using the mapping

 .

The mapping *c* is not necessarily surjective. The range of *c* determines the subset of *Al*  available for exploration by a GA.

The range of *c*, *Ξ*

 

is needed to account for the fact that some strings in the image *Al* under *c* may represent invalid solutions to the original problem.

The string length *l* depends on the dimensions of both *H* and *A,* with the elements of the string corresponding to *genes* and the values to *alleles*. This statement of genes and alleles is often referred to as genotype-phenotype mapping.

Given the statements above, the optimization becomes:

 ,

given the function

 .

Finally, with GAs it is helpful if *c* is a bijection. The important property of bijections as they apply to GAs is that bijections have an inverse, i.e., there is a unique vector *x* for every string and a unique string for each *x*.

1. **Genetic algorithms and their operators**

The operators by which GAs search for the optimal solution are set out in the following statements from Coello *et al*. (Coello, 2002) (their notation is used with some slight modifications):

Let *H* be a nonempty set (the individual or search space),  a sequence in  (the parent populations),  a sequence in (the offspring population sizes),  a fitness function,  {true, false} (the termination criteria), {true, false}, *r* a sequence  of recombination operators *τ(i)* : , *m* a sequence of *{m(i)}* of mutation operators in *mi*, , *s* a sequence of *{si}* selection operators *s(i)*: ,  (the recombination parameters),  (the mutation parameters), and  (the selection parameters).

 Coello *et al*. (2002) define the collection *μ* (the number of individuals) via *Hμ*. The population transforms are denoted by, where. However, some GA methods generate populations whose size is not equal to their predecessors’. In a more general framework can accommodate populations that contain the same or different individuals. This mapping has the ability to represent all population sizes, evolutionary operators, and parameters as sequences.

The execution of a GA typically begins by randomly sampling with replacement from *Al.* The resulting collection is the initial population, denoted by *P(0)*. In general, a population is a collection of individuals, where, and populations are treated as n-tuples of individuals. The number of individuals (*μ)* is defined as the population size.

The work of Lamont and Merkle (Lamont, 1997) defines the termination criteria and the other evolutionary operators (EOs) in more detail.

Since *H* is a nonempty set,, and, the fitness scaling function can be defined as and a related fitness function as. In this definition it is understood that the objective function *f* is determined by the application, while the specification of the decoding function *c[[1]](#footnote-1)* and the fitness scaling function *Ts* are design issues[[2]](#footnote-2).

Execution of a GA typically begins by randomly sampling with replacement from *Al*. The resulting collection is the initial population, denoted with *P*. More generally, a population is a collection *P = {a1 ,…,aμ* } of individuals . Again, the number of individuals (*μ)* is referred to as the population size.

Following initialization, execution proceeds iteratively. Each iteration consists of an application of one or more EOs. The combined effect of the EOs applied in a particular generation  is to transform the current population *P(t)* into a new population *P(t+1)*.

In the population transformation (the parent and offspring population sizes, respectively). A mapping  is called a population transformation (PT). If, then *P* is a parent population and *P/* is the offspring population. If, then it is called simply the population size.

The PT resulting from an EO often depends on the outcome of a random experiment. In Lamont and Merkle (1997) this result is referred to as a random population transformation (RPT or random PT). To define RPT, let and  be a set (the sample space). A random function  is called an RPT. The distribution of PTs resulting from the application of an EO depends on the operator parameters; in other words, an EO maps its parameters to an RPT.

Now that both the fitness function and RPT have been defined, the EO can be defined in general: let, *X* be a set (the parameter space) and  a set. The mapping



is an EO. The set of EOs is denoted as.

There are three common EOs: recombination, mutation, and selection. These three operators are roughly analogous to their similarly named counterparts in genetics. The application of them in GAs is strictly Darwin-like in nature, i.e., “survival of the fittest.”

In defining the recombination operator Lamont and Merkle (1997) let. If there exists, and , such that one individual in the offspring population depends on more than one individual of *P,* then *r* is referred to as a recombination operator.

A mutation is defined in the following manner: let . If for every, for every , for every , and if each individual in the offspring population  depends on at most one individual of *P,* then *m* is called a mutation operator.

Finally, for a selection operator: let. If,,in all cases, and *s* satisfies , then *s* is a selection operator.

1. **How genetic algorithms work**

GAs are best understood as metaheuristics, with the primary use of solving optimization problems–in particular hard combinatorial optimization problems. In the spirit of that use the following is a description of how GAs work.

Optimization theory states that if there are global minima, there needs to be a unique convex hull. If the hull exists, a simplex[[3]](#footnote-3) can be formed and a solution generated. There are methods of solving the convexity problem such as the Nelder-Mead method and others, but here the simplex stands in for these methodologies.

A convex polyhedral set is defined in the following manner:

 ,

i.e., there exists a convex hull [*conv(P)*], a nonempty finite set of vectors , and a finitely generated cone *C*.

In piecewise linear topology (PL-topology, *see the related section in this Encyclopedia*) two of the four categories are *F* and *P,* where the objects of *F* are the t-polyhedra[[4]](#footnote-4) and its morphisms (continuous mappings) and the objects of *P* are the polyhedra and their morphisms (the piecewise linear mappings–PL-mappings). PL-mappings linearly transform the convex polytopes of some covering of the domain into polytopes of some covering of the range.

In optimization theory the inequalities or constraints that define the facets of the feasible set (the subsets of the search space *H* in GAs) are characterized by polyhedrons that are elements of *P*. The feasible set is defined as nonempty and compact. In general, the feasible set is described as the intersection of the *m* closed half-spaces, i.e., convex polytopes. In both deterministic optimization and GAs the functionsand  exist. And for both optimization techniques there are simplex-wise linear embeddings of the triangulation into Euclidean space.

Simplex-wise is appropriate here because if each constraint in an optimization problem defines a half-space, the feasible set formed by this intersection of half-spaces is a simplex. *K*, the third category in PL-topology, is (are) the simplicial mappings or the PL-mappings that linearly transform each simplex of the domain onto the mappings of the range. The subsets of the vertices of *K* are the feasible sets generated by the GA.

How the feasible sets are generated by GAs is described in Radcliffe (Radcliffe N. , 1992) and in Radcliffe and Surry (Radcliffe P. a., 1996). A problem domain *f* consists of a set of problem instances *Al*, each of which takes the form of a search space(of candidate solutions) *H* together with some fitness function defined on that search space. A characterizationof the domain specifies a set of equivalences among the solutions for any instance *Al*. These equivalences induce a representationmade up of a representation space(*chromosome* in GA terminology) and a growth function *c* mapping chromosomes to the objects in *H*. The chromosome *s* is a string of alleles, which indicates that *s* satisfies a particular equivalence on *H*.

The Radcliffe and Surry and Radcliffe characterization generates a formal representation (representation space and growth function) for any instance of the problem by defining a number of equivalences over the search space. These equivalences induce subsets of the search space that contain solutions, possibly as partitions generated by equivalence relations or simply as groups of solutions sharing some characteristic. For a given solution the pattern of its membership in the specified subsets is used to define its alleles and possibly its genes. Although in some problems the search space can be partitioned orthogonally(meaning that all combinations of alleles represent valid solutions), this is not always the case[[5]](#footnote-5).

Since GAs can generate convex hulls (the facets of the feasible set) and have been shown to be simplex-wise linear embeddings of the triangulation into Euclidean space, we use the work of Alboul and van Damme (Alboul, 1997) and others to show that GAs can find the unique convex hull.

 Alboul and van Damme consider an objective function that is a discrete measure of the *L1*-norm of the Gaussian curvature over a triangle mesh. This function has a very important property. As proven by Alboul and van Damme , the use of a local edge-swapping algorithm leads to the problem’s unique global minimum, which corresponds to the unique convex hull. As noted by Effenberger (Effenberger, 2011), this measure (referred to in the literature as *tightness*) is a topological condition, meaning that any simplex-wise linear embedding of the triangulation into Euclidean space is as convex as possible. The measure can be understood as a generalization of the concept of convexity. Effenberger proves that with regard to PL-embeddings of PL- manifolds, the tightness of combinatorial manifolds can be defined in a purely combinatorial way:

(i) A simplicial complex *K* that has a topological manifold as its underlying set *|K|* is called a triangulated manifold. *K* is called a combinatorial manifold of dimension *d* if all vertex links of *K* are PL *(d−1)* spheres, where a PL *(d −1)* sphere is a triangulation of the *(d −1)* sphere that carries a standard PL structure.

(ii) Let *G* be a field. A combinatorial manifold *K* on *n* vertices is called *(k-1)* tight with respect to *G* if its canonical embedding  is *(k-1)* tight with respect to *G*, where denotes the *(n−1)*-dimensional simplex.

Prestifilippo and Sprave (Prestifilippo, 1997)as well as Weinert *et al.* (Weinert, 2003) have shown that GAs can replicate Alboul and van Damme’s edge-swapping algorithm, so by definition GAs can find the unique convex hull.

# References

Alboul, L. a. (1997). Polyhedral Metrics in Surface Reconstruction: Tight Triangulations. In T. A. Goodman, *The Mathematics of Surfaces VII* (pp. 309-336). Midsomer Norton: The Institute of Mathematics and Its Applications.

Coello, C. V. a. (2002). *Evolutionary Algorithms for Solving Multiobjective Problems.* Kluwer Academic.

Effenberger, F. (2011). Stacked Polytopes and the Tight Triangulation of Manifolds. *Journal of Combinatorial Theory, Series A* *, 18* (6).

Lamont, L. a. (1997). A Random Based Framework for Evolutionary Algorithms. In T. Back, *Proceedings of the Seventh International Conference on Genetic Algorithms.* Morgan Kauffamn.

Prestifilippo, G. a. (1997). Optimal Triangulation by Means of Evolutionary Algorithms. *Conference Proceedings: Genetic Algorithms in Engineering* (p. 492). Glasgow.

Radcliffe, N. (1992). Equivalence Class Analysis of Genetic Algorithms. *5* (2).

Radcliffe, P. a. (1996). Formal Algorithms + Formal Representations = Search Strategies. *LNCS* (1141).

Weinert, K. M. (2003). New Solutions for Surface Reconstruction From Discrete Point Data by Means of Computational Intelligence. In M. A. Wilson, *Mathematics of Surfaces: 10th IMA International Conference* (pp. 431-438). New York: Springer.

1. Remember that if the domain of *c* is total, i.e., the domain of *c* is all of *A I*, *c* is called a decoding function. The mapping of *c* is not necessarily surjective. The range of *c* determines the subset of *Al* available for exploration by the evolutionary algorithm. [↑](#footnote-ref-1)
2. The “best” fitness scaling functions will be determined in the next section. [↑](#footnote-ref-2)
3. A simplex is defined as the convex hull of a set of *n+1* points, where *n* is the number of variables. [↑](#footnote-ref-3)
4. A space homeomorphic to a polyhedron is called a topological polyhedron. This space is often referred to as a t-polyhedra. [↑](#footnote-ref-4)
5. To strengthen the tie between GA-operation and PL-topology equivalence relations exist in piecewise linear topology as they do in GAs. In PL-topology an equivalence relation is defined as a relation that partitions a set so that every element of the set is a member of one and only one cell of the partition referred to above. This definition is equivalent to Radcliffe’s and Radcliffe and Surry’s, where for any instance of the problem a representation space and a growth function define a number of equivalences over the search space. [↑](#footnote-ref-5)