1

Setting the Stage for Structured Populations

The aim of this initial chapter is to introduce some concepts related to structured populations in such a way that they are seen in the more general context of graphs or networks. With only a small cost in mathematical notation, this will allow us to discuss many apparently distinct features of populations under the umbrella of an existing and well-established mathematical notation.

Structured populations are just populations in which any given individual has its own neighborhood, which is smaller, sometimes much smaller, than the size of the population. In other words, instead of all the other individuals in the population being consirered as potential mates as in *panmictic* populations, only those that are in the same neighborhood can interact. Although this "isolation by distance" is often associated with geographical separation, this is not strictly required in evolutionary-algorithm (EA) models, where only the "relational aspect" matters. In fact, there are many examples of biological niches and isolated or semi-isolated populations in biology in which physical distance is the key factor keeping these *demes* nearly independent of each other. And many such biologically inspired models have been proposed for EAs. However, what counts is the neighborhood relationship, and this can be of any type, as long as it makes algorithmic sense. We shall see many examples of this in the following chapters. Thus, we are led to the conclusion that the important idea is the ensemble of relations among individuals, be they truly spatial or not. The mathematical objects that are required for describing this state of affairs are *graphs*. This means that we do not always need the concept of a metric space and an associated distance, such as the Euclidean distance. More often, distances between individuals will be given by the network itself, as measured along the path that links the two individuals in the graph. In these *relational graphs* the rules governing the construction of the graph do not depend upon any external metric between the vertices. For example, the electrical supply network of a given region would be a kind of spatial graph, since the distances between the graph nodes (generators, transformers and so on) are relevant, while the network of acquaintances in a society conveys only relationships, although other kinds of non-Euclidean

distances can be associated with such networks. In the realm of population graphs, and with a slight abuse of language, I shall often call these population structures *topologies*, be they truly spatial or not.

Since graphs are a suitable mathematical description for structured populations, I shall give here a short introduction to the relevant concepts and definitions that will be used in the rest of the book. Graph theory is a well-developed branch of discrete mathematics and it would be impossible, and also useless, to try to give an account of it here, however brief. Instead, I shall limit myself to the introduction of the concepts that are really useful to us. In Chap. 6, we shall need a few more ideas about graphs. Since these concepts are not needed yet, I shall defer their presentation to the relevant place.

1.1 Useful Definitions for Graphs

For ease of reference, I collect here a few definitions and some nomenclature for graphs that will be used throughout this work. The treatment is necessarily brief: a more detailed account can be found, for example, in [22].

Let V be a nonempty set called the set of *vertices* or *nodes*, and let E be a symmetric binary relation on V, i.e. a set of unordered pairs of vertices. $G = (E, V)$ is called an *undirected graph*, and E is the set of *edges* or *links* of G. In a *directed graph* edges have a direction, i.e. they go from one vertex to another, and the pairs of vertices are ordered pairs. Figure 1.1 shows an undirected and a directed graph.

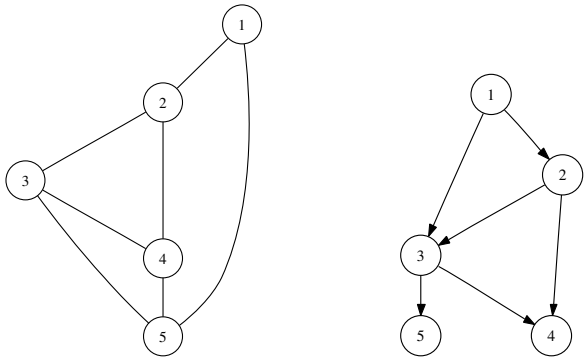

Fig. 1.1. (a) An undirected graph. (b) A directed graph

A *subgraph* of G is a subset of a graph's edges and associated vertices that constitutes a graph. That is, $G^{'} = (V^{'}, E^{'})$ is a subgraph of $G = (E, V)$ if $V^{'} \subseteq V$ and $E' \subseteq E$. For example, the set of vertices $1, 2, 4, 5$ in the graph G shown in Fig. 1.1 a induces a subgraph of G.

When vertices (u, v) of an undirected graph G form an edge, they are said to be *adjacent* or *neighbors*. The *neighborhood* of a vertex v is the set of vertices that are adjacent to v in G, not including v. For example, the neighbors of vertex 4 in Fig. 1.1 (a) are 2, 3, and 5. The *degree* k of a vertex is the number of edges impinging on it (or, equivalently, the number of neighbors). For directed graphs, one can correspondingly define the *outdegree* of a vertex v, which is the number of edges that leave v, and the *indegree*, which is the number of edges that enter the vertex v. The adjacency relation is not symmetric for directed graphs. For example, vertex 1 in the graph in Fig. 1.1 b has an outdegree of 2 and an indegree of 0.

A *path* from vertex u to vertex v in a graph G is a sequence of edges that are traversed when going from u to v with no edge traversed more than once. The *length* of a path is the number of edges in it. For example, the sequence $\langle 1, 2, 3, 5, 4\rangle$ is a path from vertex 1 to vertex 4. The *shortest path* between two vertices u and v is the path with the smallest length joining u to v. Thus, in Fig. 1.1 a, the sequences $\langle 1, 5, 4\rangle$ and $\langle 1, 2, 4\rangle$ are the two shortest paths from node 1 to node 4. Edges can have weights associated with them in some applications, and these weights are used in the calculation of the path lengths. If nothing is stated, each edge has a unit weight.

Cyclic paths are particular paths in a graph whose first and last vertices are the same. A *tour* is a particular cycle that contains every vertex. Again referring to Fig. 1.1 a, $\langle 1, 2, 4, 5, 1\rangle$ is a cycle, while $\langle 1, 2, 3, 4, 5, 1\rangle$ is a tour.

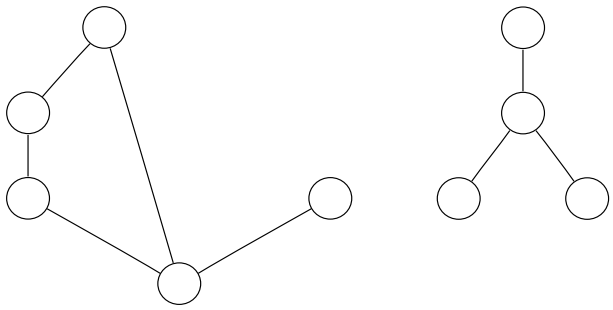

Fig. 1.2. An unconnected graph with two connected components

The maximum distance (path length) between any two connected vertices of a graph is called the *diameter* of the graph.

A graph is *connected* if there is a path between any two vertices. A graph that is not connected consists of a set of *connected components*, as illustrated in Fig. 1.2.

A *completely connected* undirected graph G with $|V| = N$ vertices has an edge between any two vertices. The total number of edges is thus $N(N-1)/2$. Figure 1.3 shows an example of such a graph with $N = 5$.

A *clique* in an undirected graph G is a completely connected subgraph of G.

A *sparse* graph has a number of edges $|E| \ll N(N-1)/2$. A *dense* graph has a number of edges $\propto N^2$.

A *star graph* is a network in which there is a particular node (the center) that is connected to all the other nodes, while the rest of the nodes are connected only to the center (Fig. 1.4).

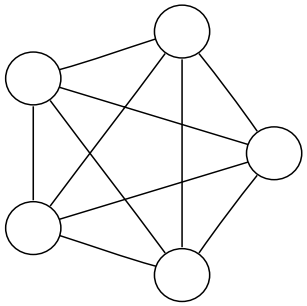

Fig. 1.3. A complete graph with five vertices

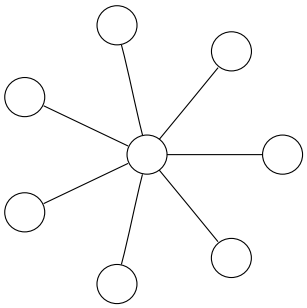

Fig. 1.4. A star graph

Finally, an *hypergraph* is like an undirected graph, but each edge connects an arbitrary subset of vertices and is called a *hyperedge*.

A graph G in which all the vertices have the same degree k is called *k-regular*. Complete graphs are obviously regular. Another important class of regular graphs for us is the family of *lattice graphs*. A *d-lattice* (or d-dimensional lattice) is an unweighted, undirected graph in which each vertex has the same degree k, with $k \geq 2$ and $k \geq 2d$. For example, with $d = 1$ and $k = 2$ and with $d = 1$ and $k = 4$, one obtains the ring structures shown in Fig. 1.5.

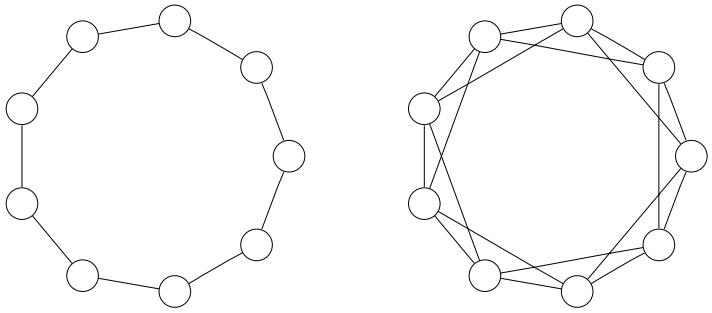

Fig. 1.5. (a) one-dimensional lattice with $k = 2$. (b) one-dimensional lattice with $k = 4$. Periodic boundary conditions are assumed

With $d = 2$ and $k = 4$ we have a torus, which is a topological entity with periodic boundaries (see Fig. 1.6).

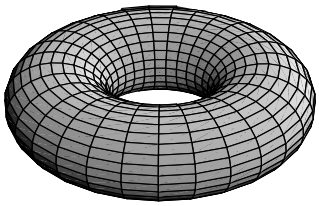

Fig. 1.6. A torus topology. This is obtained by wrapping the rows and columns of a two-dimensional grid around on themselves

Some of these graph types have been often used in evolutionary computation, and the dynamical properties of populations structured in these ways will be studied in detail in Chaps. 4 and 5.

1.2 Main Graph Structures of Populations

I shall now introduce the principal structures of populations that have been used in EAs, and these will be the base types for our analysis. Here I shall discuss only their graph-like properties. The dynamical evolutionary processes taking place in the populations will form the main theme of the book and will be dealt with in detail in the following chapters. Networks can be considered

as the backbone on which dynamical processes take place. Therefore, networks are a prerequisite for describing the behavior of complex systems. Of course, the static and dynamic views are not divorced in reality, since there are mutual interactions between them. Think, for instance, of the communication processes taking place on a network that is itself continually changing, such as the Internet. However, our artificial structures are simpler, and it is useful to single out and consider their static structure first.

1.2.1 Island or Multipopulation Models

Here the idea is simply to divide a large panmictic population into several smaller ones. This model is usually called the *island* model or *multipopulation model*, and is schematically illustrated in Fig. 1.7. Each subpopulation runs a standard sequential EA, and individuals are allowed to migrate between populations with a given frequency. The migration directions are represented in the figure by arrows.

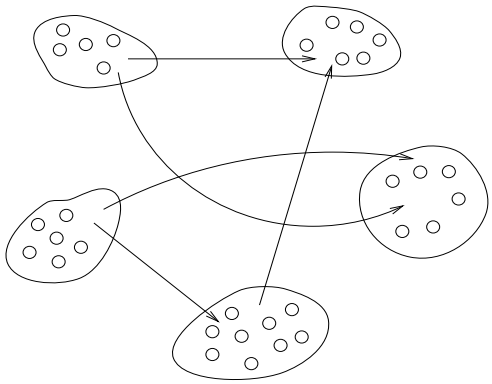

Fig. 1.7. A multipopulation structured model. Each "blob" represents a panmictic subpopulation. Subpopulations are loosely connected by periodically sending and receiving individuals according to the pattern shown by the arrows

In graph theory terms, each subpopulation is a vertex of the graph, and the edges are given by the migration links between islands. Note that the graph is usually a directed one. At a lower level, each island could be seen virtually, in turn, as containing a population structured as a complete graph. Several patterns of connection have traditionally been used. The most common ones are rings, two-dimensional and three-dimensional lattices, stars, and hypercubes. We shall deal with these models at length in Chaps. 2 and 3.

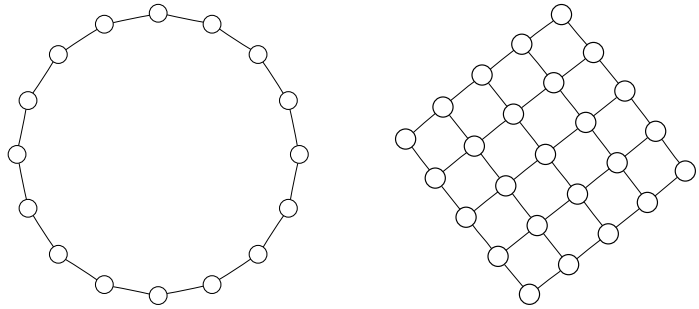

Fig. 1.8. (a) A one-dimensional-ring cellular population structure. (b) A two-dimensional-grid cellular population structure. In both cases each node is a single individual, and the edges wrap around in the grid case

1.2.2 Cellular Models

In *cellular* models, also called *diffusion* models, the individuals making up the population are usually disposed according to a regular lattice topology, i.e. a lattice graph (see Sect. 1.1). Two examples in one dimension and two dimensions, respectively, can be seen in Fig. 1.8, where the different shapes of the nodes represent potentially distinct individuals. In graph theory terms, each individual is a vertex of the graph, and edges link adjacent individuals, i.e. neighbors. In these cellular populations, each individual interacts only with a few other individuals in its neighborhood, and all genetic operations are local. Lattice-graph cellular populations will be examined in detail in Chap. 4. Of course, we are by no means limited to cellular models that are mapped onto regular lattices, although this has been the rule in the EA world. We shall see in Chap. 6 that many other graph topologies are possible and useful, including random and irregular structures.

Finally, it is worthwhile to make a comment on an interesting proposal of Sprave [141], who suggested using the hypergraph formalism to describe structured EA populations. Hypergraphs (see Sect. 1.1) are a generalization of simple graphs in which edges may span a subset of vertices. Hypergraphs can thus model any population structure, including the limiting panmictic case. Sprave's suggestion is attractive because it allows an elegant unified description of structured populations. However, for the sake of simplicity, and also because some new graph structures (see Chap. 6) are rather clumsy to represent with hypergraphs, I have chosen to stick with the more standard view of simple graphs.

1.2.3 Other Topologies

It is of course possible to design and implement population topologies more complex than the "basic" types described above. For example, one could have a multipopulation structure in which each subpopulation has a cellular topology. Alternatively, it would be possible to have a hierarchical island system, in which each island at an upper level contains a number of islands at a lower level. Exchanges would then be limited to taking place between islands at the same level.

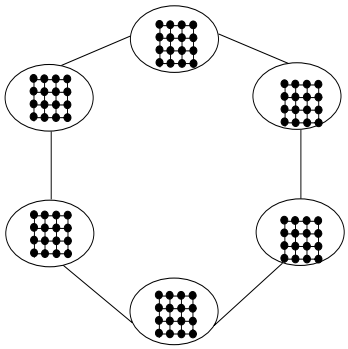

Fig. 1.9. A hierarchical EA model in which the populations in the loosely connected islands have a lattice structure.

Several other possibilities spring to mind as well. Figures 1.9 and 1.10 provide a schematic view of two *hierarchical*, or hybrid, population structures such as those described above.

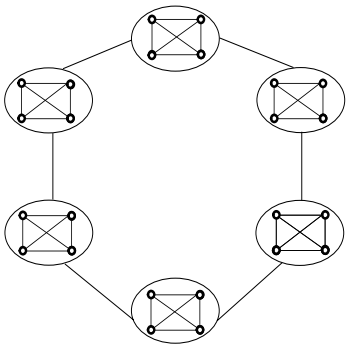

Fig. 1.10. A hierarchical EA model in which each island at the outer level contains a multipopulation EA. The communications between islands at the inner level are more frequent than the migrations at the outer-island level

Some of these structures, or analogous ones, have indeed been used in empirical work with good results. It would be tempting to try to extend the analysis that we shall do for the simpler cases to these more complex situations. However, I feel that our tools are not yet sharp enough to successfully deal with these more difficult cases (note than an analysis of a hierarchical master–slave EA model has been presented by Cantú-Paz in Chap. 8 of his book [24]). On the other hand, the simpler topologies already offer an extremely rich behavior and constitute a large field of investigation in themselves. As a consequence, I shall limit myself in the following to the description and analysis of "pure" island and cellularly structured population models.