1 Introduction

In solid mechanics, a *structure* is defined as a connected domain of material(s) (i.e., a body) capable of transferring loads from some point(s) of application to some other point(s) of fixity. These points of application and fixity, and indeed the limits of the structural domain itself, are defined by the scope and scale of the application being designed and its relationship to the engineer or designer. For example, during preliminary layout design an entire aircraft wing might be viewed as a structure, while later in the detailed design process a single rib might be carefully configured as a distinct structural component. Finally, those developing advanced materials might consider the distribution of voids and properties at the material level to represent a problem of "microstructure."

In engineering terms, all structures that can be designed in 2-D/3-D space have exterior boundaries, and, regardless of scale, the relative spatial placement of material(s) and voids within these boundaries defines a structure's **topology**. Mathematically, two objects are topologically equivalent if each can be continuously deformed into the other. Thus, a sphere in three dimensions is topologically equivalent to an ellipsoid but not to a doughnut, given that a hole would have to be cut in the sphere in the latter case. The determination of an appropriate, or even optimal, topology is typically one of the first steps toward developing a structural design. The goal of this book is to expose the reader to techniques that enable expedient design space exploration to identify novel, optimal topologies.

Once a topology has been determined, two secondary but important classes of structural design decision must be addressed, the most common motivations for which are simplification of engineering representation and definition, as for eventual fabrication. **Shape design** in structures refers to the refinement of topology to consider 2-D and 3-D shapes, usually of some canonical type, assigned to particular regions of a structure. Returning to the mathematical definition above, topological equivalence partitions a set of objects into disjoint sets of shape classes, with each member of a shape class being a continuous deformation of any object of the same shape class. Therefore, in shape design, the optimization is restricted to remain inside the given shape class and is thus greatly limited when compared with topology optimization. **Sizing design** is the subsequent process of assigning specific dimensions to these shapes (e.g., radii of holes and fillets, thickness of features). Therefore, in sizing, only a reduced set of deformations is allowed, further limiting the scope of the structural optimization.

Shape and sizing design are of course critical to the full design of a structure and will be addressed throughout this book beginning in Section 1.2.4 and much more rigorously in Chapters 2 and 4 (Sections 2.3 and 4.4.1, respectively). Further, the topology, shape, and sizing design phases are also addressed in other resources (Bendsøe & Sigmund 2003, Querin, Victoria, Alonso, Loyola & Montrull 2017). However, in this work we focus primarily on that first and most essential challenge: the selection and representation of topology.

1.1 Structural Topology and Representation

The aim of this overall work is to develop and demonstrate developmental/algorithmic approaches for structural representation and design that leverage biological analogies, allowing optimized structures and networks to be developed and evolved. Clearly, the problem of topological design is not limited to the engineering of solid bodies for the carrying of mechanical loads; all networks have as a fundamental property their topology.[1] Communication networks, electrical circuits, and fluidic and chemical processes all rely on selective connectivity between nodes to perform their intended functions. Topological design/optimization is the process of determining/optimizing these nodes and connectivities. Our goal is then to provide engineers and designers with a new set of tools for optimizing topologies across all topics of engineering (solid mechanics, electronics, communications, fluidics, etc.). Indeed, it will also be shown that extensions to the graphical and environmental design arts are also possible.

From design to physical realization, all bodies and networks subject to analysis and/or intended for precise replication must be deterministically represented. In illustrating such representation, we consider, in particular, the definition of solid bodies (e.g., those occupying nonzero two-dimensional area or three-dimensional volume) as both a general and a motivating case. Such a task is general in that it requires the consideration of faces or volumes, while the definition of networks and circuits requires the consideration of only nodes and edges. The task is motivating inasmuch as it will be seen to represent the majority of examples provided throughout this text. The topological and geometric representations that will be addressed throughout this work are illustrated for a two-dimensional domain in Figure 1.1.

A deterministic and quantitatively precise representation of a body allows a finite array of design variables to define the configuration of that body in such a way that (i) the effect of design changes on engineering performance can be assessed and (ii) the body may eventually be realized as a prototype or product. As shown in Figure 1.1,

[1] Throughout this work, a *structure* is defined as an entity composed of material and having nonzero mass and volume (though it may be represented as two-dimensional). Therefore, it is fundamentally characterized by its *geometric* features (e.g., lengths, angles, areas, volumes). A *network* (or *graph*) is purely *combinatorial*, consisting of nodes and their connections, where the concepts of geometry are fundamentally irrelevant to functional performance. Note that networks, if assigned geometric properties, can be interpreted as structures. The fact that topological representation is common to both structures and networks is central to this body of work, which addresses both.

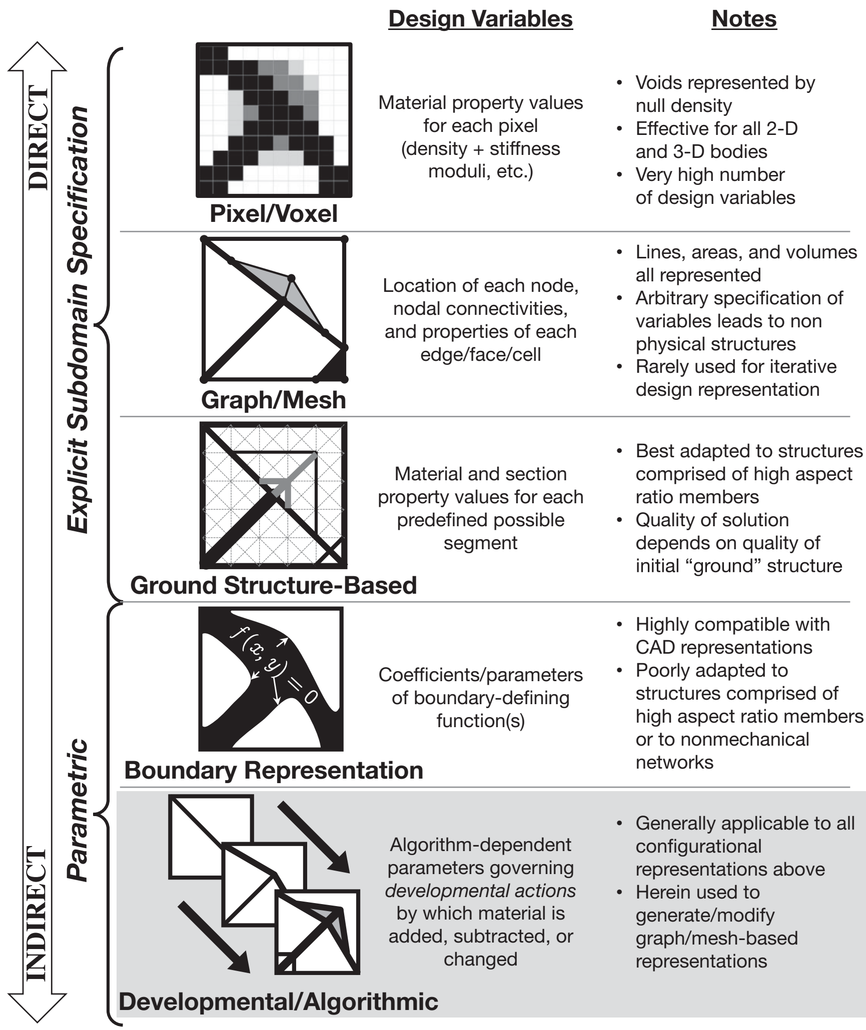


Figure 1.1 Approaches to structural and/or network representation used in engineering and design processes. Representations range from physical to abstract based on the extent to which design variables explicitly determine the spatial location of material and void regions. This work focuses on new *developmental* approaches using bioinspired algorithms.

such representations can be more *physical* in that design variables directly define the locations of material(s) and void(s), or they can be more *abstract*, in that no connection between the design variables and the resulting connected domain can be explicitly observed or intuited. Specifically, two classes of representation are proposed:[2]

[2] In the topological modeling community, only three representation options are often considered: *pixel/voxel*, *graph/mesh*, and *boundary representation*. While *ground structure-based* and *developmental/algorithmic* approaches might be considered as derivative of the other three, they are considered herein for the distinct and effective manner in which design variables are linked to geometric and topological configurations via these representations, especially for the purposes of iterative design.

https://doi.org/10.1017/9781009026635.001 Published online by Cambridge University Press

- **Direct Subdomain Specification:** In these representations, at least some subset of the internal variables is 1-to-1 mapped to the actual spatial location of portions of the body (small material region, segment/ligament, face, etc.). Other associated design variables might be used to further define the properties of each subdomain.
  - *Pixel/Voxel* representations are the most straightforward of all options in that they directly specify the type and properties of each material at each point in the domain, subject to the level of refinement of discretization over a fixed regular grid. Voids are defined simply as the absence of material. This ultimate generality comes at the cost of a very high number of design variables, the physical limit being the continuum hypothesis.
  - *Graph/Mesh* representations consider arbitrary and spatially disparate discretizations of solid domains into volumetric cells or two-dimensional faces bounded by edges, the intersections of which are defined by nodes. Some design variables then directly specify the location of nodes and their connectivity into the aforementioned subdomains, while others specify the properties of materials within these subdomains. Because graphs are not guaranteed to be physically realizable (e.g., overlapping domains are mathematically permitted) and because the underlying data structure of the design variables can by highly complex for arbitrary bodies, this approach is rarely used for iterative configurational design, though it can be highly precise for physical realization once a configuration has been otherwise optimized.
  - *Ground Structure-Based* representations combine the features of pixel/voxel and graph/mesh representations in that they define graphs over fixed regular grids. They are most commonly used in the definition of structures comprised of high aspect ratio segments or of networks such as circuits, which are entirely edge-based. Design variables are similar to those in a pixel/voxel representation, defining which segments are assigned properties and which are not (and are thus nonexistent).
- **Parametric:** In these representations, none of the design variables can be explicitly related to the spatial location of any region of material. Rather, the variables are used as parameters in mathematical functions or algorithmic processes that then determine where material is, where it is not, and what material properties should be considered.
  - *Boundary Representations* define a body by the divisions between material and void, not by the spatial distribution of material explicitly. In general, this may be accomplished through any surface representation,[3] though in the context of shape modeling, computer-aided drafting (CAD), and the iterative design considered herein, this is usually accomplished through the use of mathematical functions capable of capturing linear/planar, cylindrical, and conic features, in addition to more generalized features through the use of such functions as nonuniform rational basis splines (NURBS). Because these approaches are specifically

[3] See, for example, the ubiquitous use of `.stl` files in additive manufacturing, which use triangular meshes to define closed surfaces, indicating where the material is and is not to be deposited.

formulated to define domains of nonzero area and volume, they suffer when local aspect ratios of regions become very high and are entirely inappropriate for the representation of edge-based networks.

– *Developmental/Algorithmic* representations are by far the most abstract in that the design variables, while fully sufficient to deterministically define a design configuration, are in no way linked directly to any observable geometric or topological feature of a body or network. Rather, they are explicitly used as *instructions* to an algorithm that creates or modifies any of the representations described above. The final configuration is then defined by the design variables as they inform cyclic developmental processes, each cycle usually increasing the complexity of the configuration.

1.2 Topology Optimization Toward Preliminary Structural Design

A well-chosen and well-developed structure or network representation allows for interdisciplinary technical communication, tractable quantitative design iteration, and ultimately the fabrication of systems and solutions. Of these, this work specifically focuses on the design implications of structural representation, especially as one seeks to determine optimal topologies in a rigorous, analysis-driven manner.

1.2.1 The Structural/Network Design Problem

The process of topological structure design is the means by which one answers the following motivating question:[4]

> How should one best configure material(s) into a geometric form to perform a needed *function* or *functions* subject to some *constraints* and given some metric for ranking design preferences, known as *objectives*?

These three design concepts are essential for rigorously posing and efficiently solving the structure/network design problem and thus should be clearly defined.

- **Functions** define the purpose of a structure or network, sometimes in the context of a larger system. In mechanical structures, this usually involves the transfer of loads from one or more points of application to one or more points of fixity. In compliant mechanisms, it may involve the transfer of displacements; in other networks, it usually involves the transfer of information between multiple points. Multifunctional structures might be tasked with one or more of the above.

[4] Alternatively, the process of *network* topological design is the process of answering the question “How should one best configure the connectivity between a (sometimes modifiable) set of nodes, and what should the properties of those connections be, to perform needed functions subject to some constraints and given some objectives?”

- **Constraints** are limits on global or local failure and/or gross dimensional restrictions and fixed geometric definitions (e.g., the particular locations of load application) that define the feasibility of a solution. They may also include limits on deformation in mechanical structures or on the loss of information or energy in other networks.
- **Objectives** are the goals that allow preferential ranking of designs, such as minimization of cost or of mass, minimization of manufacturing complexity, maximization of service life, and so on. Strict constraints can also be recast as less strict preferences in some cases, resulting in additional objectives, such as the minimization of deflection under load.

To answer our motivating question for a given functional need, an engineer or designer might take a number of approaches. Let us examine the following simple example.

Example 1.1 Consider the design of a simple structure intended to carry a shear load, as illustrated in Figure 1.2, where the function, constraints, and objectives associated with such a design can be clearly stated as follows:

Function: Transfer a total shear load F across some fixed distance L;

Constraint: Total structure should be at most $L \times L$ in size and no local material failure can occur under load;

Objective: Minimize total mass of material used (i.e., mass of the total structure).

Knowing the largest allowable size and the locations of load application and structural fixity, a designer might be guided by a number of influences as the preliminary ideation process begins (Figure 1.2).

- For many problems, some degree of **simplistic intuition** can provide guidance, and in this case the simple removal of material relative to the maximum allowed might allow one to arrive at a sufficiently lightweight design capable of carrying the required load without failure. Clearly, both the quantitative description of the design and subsequent fabrication via subtractive manufacturing can be very straightforward if simple geometric features are considered;
- It is very common for a designer of any engineering system or subsystem to leverage **historical successes** in configuring new solutions; the design of new structures or networks is often strongly influenced by such legacy design;
- Some degree of **rigorous quantitative analysis** can provide clear guidance toward optimal geometric configuration. For example, dominant principal stress paths computed via finite element analysis can provide clear guidance as to where material is needed and where it is not in a structure;
- Over the past few decades, rigorous methods of **topology optimization** have been proposed, demonstrated, and validated toward the automated preliminary design of structural layouts.

https://doi.org/10.1017/9781009026635.001 Published online by Cambridge University Press

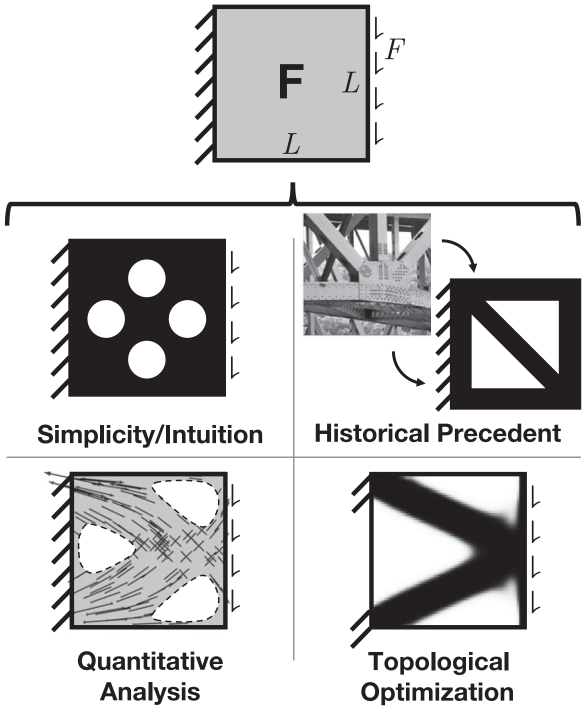


Figure 1.2 An engineer or designer may be guided by a number of different influences in configuring an optimal structure or network; this work addresses classical and novel methods of topological optimization especially for addressing design problems without historical precedent and for which intuition is lacking.

The relative strength of each influence during preliminary design will ultimately depend on such considerations as designer experience and design team capabilities, allowable design cycle cost and time, and problem complexity, among others.

As expected, the design of simple structures performing common functions has been considered across most engineering disciplines. Initial intuition and historical precedent, confirmed via quantitative analysis, are common throughout the engineering design landscape. The purpose of this book is to expand the understanding and application of the fourth option proposed: the more expansive use of topological optimization techniques throughout the engineering community. Toward that end, we next provide a brief review of the more well-known approaches, including their advantages and disadvantages. A more rigorous quantitative treatment is provided for each approach, as well as for others, in Chapter 2, but the primary intent overall is to introduce newer and sometimes much more capable bioinspired alternatives especially suited for multiobjective design problems. These problems are especially challenging as, being novel and usually highly nonlinear, they can be solved neither from intuition nor from precedent.

https://doi.org/10.1017/9781009026635.001 Published online by Cambridge University Press

1.2.2 Methods of Single-Objective Topological Optimization

The vast majority of topological optimization methods introduced in the literature and employed throughout industry consider a single-objective, which is most often the minimization of structural compliance given some constraint on the total volume. In Chapter 2, we will review the most popular methodologies for single-objective structural topology optimization while also introducing emerging techniques. With respect to the introductory concepts of this chapter, it is useful to classify these optimization methods based on the options for structural representation as shown in Figure 1.3 (cf. Figure 1.1). These are briefly summarized in the following and are described in more detail in Chapter 2 as noted.

Pixel/Voxel Representation

In general, these methods consider an initially dense domain discretized into many finite elements (Cook, Malkus & Plesha 1989, Ern & Guermond 2004, Reddy 2019)

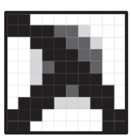
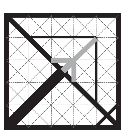
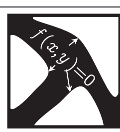
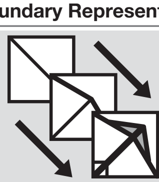
	Topology Optimization Approaches	Summary
 Pixel/Voxel	• Solid Isotropic Material with Penalization (SIMP)	Local pixel/voxel densities are varied to minimize a global objective function (e.g., a compliance functional) while satisfying global volume constraints (Section 2.1.1)
	• Evolutionary Structural Optimization (ESO)	Heuristic "hard-kill" approach, whereby individual pixels/voxels satisfying some criteria (e.g., stress below some value) are removed (Section 2.1.2)
	• Homogenization Method	Theories of micromechanics are used to directly optimize local "periodic lamina," resulting in grey material removed via postprocessing (Section 2.1.3)
	• Genetic Algorithm/SIMP Hybrid Method	The SIMP method is combined with a GA such that the best features from potentially parallelized SIMP optimizations are combined (Section 2.1.4)
 Ground Structure-Based	• Traditional Ground Structure Method (GSM)	Section properties of truss/frame members comprising a preconfigured physically realizable graph are varied to minimize a global objective function (Section 2.2.1)
	• Truss Geometry and Topological Optimization	Extension of GSM, whereby the configuration of the underlying graph (e.g., locations of nodes) is itself modified during the optimization process (Section 2.2.2)
 Boundary Representation	• Level Set Method	Functions defining the boundaries between solid material and void are modified in such a way that best minimizes a global objective function (Section 2.3.1)
 Developmental/ Algorithmic	• Turtle Graphics	Genes evolved by GA are used to define an L-system, which provides instructions to a classical "turtle" algorithm drawing branched graphs (Chapter 3)
	• Spatial Interpretation for the Development of Reconfig. Structures (SPIDRS)	Uses GA to evolve L-system toward global objectives as in Turtle, but graph-generating instructions are uniquely intended to provide effective connectivities (Chapter 4)
	• Arrangement L-system Interpretation	GA again used to evolve a rewriting language; however, recursive process is interpreted *in situ* to produce graphs via face/cell division (Chapter 5)

Figure 1.3 A range of single-objective topology optimization approaches has been proposed, each based on one of the structural representation options as introduced in Figure 1.1.

https://doi.org/10.1017/9781009026635.001 Published online by Cambridge University Press

with the goal of minimizing an objective function by identifying whether a given element should consist of solid material or void. Design variables associated with each pixel/voxel/element may have discrete “0/1” or continuous values designating local density, depending on the method employed (Bendsøe 1989, Eschenauer & Olhoff 2001). By far the most popular approach is the solid isotropic material with penalization (SIMP) method (see Section 2.2.1), which often involves the mathematical minimization of a structural compliance metric via continuous variation of local densities, these being strongly penalized when taking on values other than 0 or 1. Two other approaches are heuristic in nature and thus can consider binary density values (i.e., full material or void). These include evolutionary structural optimization (ESO) and the Lamarckian genetic algorithm/SIMP hybrid method.

Evolutionary Structural Optimization (ESO) (see Section 2.2.2) is a “hard-kill” method based on the idealization that optimal structures are made up of material regions that are all fully stressed (Xie & Steven 1997*a*). ESO incrementally removes “inefficient” material from the design domain based on heuristic criteria such as elemental stress (Xie & Steven 1997*a*) or other sensitivity values (Liang, Xie & Steven 2000). A variant known as bidirectional ESO (BESO) also allows for material to be added in regions where elements have a high criterion value (Querin, Steven & Xie 1998, Querin, Young, Steven & Xie 2000). The Lamarckian genetic algorithm/SIMP hybrid method (see Section 2.2.4) uses a genetic algorithm (GA) to combine and evolve the results of possibly parallelized SIMP optimizations. Because GAs (introduced in Section 1.3.3) are heuristic and do not require gradient information to be calculated or approximated, a wide range of objectives and constraints can be considered.

Regardless of the method employed, however, all pixel/voxel approaches suffer from important challenges. Given that each 2-D pixel or 3-D voxel is associated with its own design variable, the dimensionality of the design space increases rapidly when considering that a higher resolution of the design space (which results in more accurate simulated results) requires an increased number of pixels/voxels (Deaton & Grandhi 2014, Kobayashi 2010). This problem is greatly compounded for three-dimensional structures, for which SIMP optimization requirements in terms of computational resources and time can explode (Aage, Andreassen, Lazarov & Sigmund 2017). At the same time, density-based methods alternatively suffer from a phenomenon known as “checkerboarding,” resulting from high numerical stiffness of such patterns. This formation of adjacent solid/void elements arranged in a checkerboard pattern (Bendsøe 1989, Bourdin 2001, Deaton & Grandhi 2014), and indeed the generation of nonphysical or nonoptimal solutions, becomes prevalent at volume fractions that are prevalant in engineering practice (e.g., the $\sim 10\%$ volume fraction found in aerospace structures such as wings). These approaches are also notoriously mesh-dependent, meaning different optimal topologies can result from identical design domains of different discretization refinements. Finally, “gray” material resulting from local densities other than 0 or 1 can arise when the design conflict between mass and stiffness is weak and often requires postprocessing or “interpretation,” which may greatly reduce the gains of the optimized, nonmanufacturable solution obtained with the density-based method.

The ESO/BESO implementations have also been shown to break down for simple structural problems (Zhou & Rozvany 2001) and utilize heuristic criteria that may not relate to the objective function (Rozvany 2009). Both ESO and the Lamarckian genetic algorithm/SIMP hybrid method approaches can result in singularities in associated finite element matrices (Deaton & Grandhi 2014), which can be avoided when discrete 1/0 design variables are replaced with continuous variables, but this results in the formation of gray "transition" material between solid and void regions (Guest, Prévost & Belytschko 2004, Sigmund 2007, Guest, Asadpoure & Ha 2011, Kawamoto et al. 2011). Various *regularization* techniques have been created to prevent numerical issues and control the quality of final results, such as filtering methods (Sigmund 1997, Sigmund & Petersson 1998, Bourdin 2001, Lazarov & Sigmund 2011), length-scale constraints (Guest et al. 2004, Guest 2009, Lazarov & Wang 2017), and projection schemes (Guest et al. 2004, Sigmund 2007, Guest et al. 2011, Kawamoto et al. 2011, Wang, Lazarov & Sigmund 2011).

Ground Structure-Based

The ground structure method (GSM; see Section 2.3.2) is based on the concept of incrementally updating a pre-existing graph by computing the sensitivities of a given structural performance metric to the section properties (e.g., thickness, area, etc.) of each line segment in that graph. As such, it only considers structures consisting of slender structural members (trusses or frames). The topology of the domain is then evolved by varying sectional properties of members to such an extent that they can essentially be removed from the structure (Ben-Tal & Bendsøe 1993, Bendsøe, Ben-Tal & Zowe 1994, Lee, Mueller & Fivet 2016). While GSMs are adept at finding solutions to truss and frame problems, the quality of optimized results strongly depends on the quality and refinement of the initial graph, where meaningful refinement can quickly increase the number of design variables required as well as the computational time needed to analyze initially dense configurations (Bendsøe et al. 1994, Hagishita & Ohsaki 2009). Combining the sizing optimization of the GSM with shape optimization in the form of node placement optimization can mitigate the need for initially dense ground structures (Achtziger 2007). Similar to BESO, a method known as the growing GSM (GGSM) allows for both the addition and removal of structural members, allowing for the initial ground structure to remain relatively sparse (Hagishita & Ohsaki 2009). Further, an extension to GSM whereby the initial graph can itself be incrementally modified through changes in nodal locations is discussed in Section 2.3.3.

Boundary Representation

The single approach to topology optimization based on this approach for quantifying structural configurations discussed herein is second in popularity only to SIMP. Known as the level set method (see Section 2.4.1), it optimizes over the space of parameters defining the scalar-valued level set function that is used to define domain boundaries (i.e., the interface of solid and void) (Sethian & Wiegmann 2000, Osher & Fedkiw 2003, Wang, Wang & Guo 2003, van Dijk, Maute, Langelaar & van Keulen 2013, Deaton & Grandhi 2014). This allows for the convenient treatment of topological changes, as structural boundaries can be modified by using the physical problem and optimization conditions to control the output of the level set function. A level set function can be parameterized using finite element method (FEM) basis functions (Wang et al. 2003, Allaire, Jouve & Toader 2004, Amstutz & Andrä 2006, Xing, Wei & Wang 2010, van Dijk, Langelaar & Keulen 2012), radial basis functions (RBFs) (de Ruiter & van Keulen 2004, Wang & Wang 2006, Luo, Tong, Wang & Wang 2007, Kreissl, Pingen & Maute 2011), or Fourier series (Gomes & Suleman 2006), which determine the design freedom and precision of the material boundaries.

One challenge this approach faces is that performance analysis requires the mapping of a parameterized level set function to a structural model, which can impact the accuracy of the structural response and the quality of the final result of the optimization process. Currently utilized methods for this mapping procedure include conforming discretization (unique remeshing; most accurate but computationally expensive) (Ha & Cho 2008, Allaire, Dapogny & Frey 2011), immersed boundary techniques (extended FEA approaches; accurate but difficult to implement) (Kreissl et al. 2011, van Dijk et al. 2012), and density-based mapping (mapping of boundaries onto pixel/voxel representation; accurate with regularization techniques) (Wang et al. 2003, Allaire et al. 2004, de Ruiter & van Keulen 2004). Further, the level set method suffers from a poor rate of convergence, convergence to local minima, and difficulties in dealing with constraints (van Dijk et al. 2013), some of which can be prevented by using regularization techniques such as sensitivity and density filtering (Luo, Wang, Wang & Wei 2008, van Dijk et al. 2012, Zhu, Zhang & Fatikow 2015). Finally, as mentioned in Section 1.1, this approach can be ineffective as local aspect ratios become very high, as in structures composed of slender truss or frame components.

Developmental/Algorithmic

These approaches to topological design optimization are among the newest and comprise the primary topic of this work. They employ a number of different algorithms to develop a graph, and thus a structure, by executing generative actions based on instructions associated with a rewriting language. The inputs to the language then comprise the design variables of the optimization problem such that an algorithm seeks to find a set of instructions that generates the best-performing structure (or network) depending on functional goals. Turtle graphics, the simplest of these approaches, is used to introduce all critical algorithmic concepts in Chapter 3. The spatial interpretation for the development of reconfigurable structures (SPIDRS) algorithm generates much better-performing networks and is described in Chapter 4. Finally, arrangement L-system interpretations, which also generate high-performance solutions, are developed in Chapter 5.

All the various single-objective methods of topological optimization summarized above can be classified as *gradient-based* and *gradient-free* depending on whether they employ an explicit dependence on the derivatives of the objective function considered. SIMP, ground structure, and level set methods are all gradient-based

https://doi.org/10.1017/9781009026635.001 Published online by Cambridge University Press

methods, meaning that they can be numerically efficient but may have difficulties with problems containing multiple local optima, discontinuous design spaces, or discrete variables (Deb 1999). They also can explicitly consider a single-scalar-objective (and thus its derivatives) at a time. On the other hand, ESO, GA bitmap methods, and the three developmental algorithms are all gradient-free, specifically relying on metaheuristics to guide the search for better designs, while not being mathematically limited to a single scalar objective. This may require a large number of structural analyses compared to gradient-based methods, but it makes these methods robust against local optima or discontinuous design spaces. The choice of gradient-based or gradient-free algorithm will also have critical implications as one extends to consider multiple objectives simultaneously, as is the focus of this work and is described in the following sections.

1.2.3 Extension to Multiobjective Topological Optimization

While single-objective approaches have been and will continue to be used for a wide range of structural design problems, the optimization process is made considerably more difficult when the underlying problem has *multiple* performance objectives. These objectives are often conflicting, leading to a compromised topology if pursued simultaneously (Chen & Wu 1998). Thus, multiobjective topology optimization problems rarely possess a single "optimal" solution but instead exhibit a series of solutions that are classified as *Pareto-optimal*. To classify as such, a solution cannot be outperformed in all objectives by another design (Cohon 2003). This book, in particular, focuses on new bioinspired approaches for multiobjective topology optimization using algorithmic approaches, which are also briefly summarized in Figure 1.3 and comprise much of the remainder of this text. However, there are several existing multiobjective topology optimization approaches that should also be quickly reviewed prior to proceeding with the introduction of new algorithms.

Pareto-optimal solutions for topology optimization problems have been obtained by implementing the algorithms listed in Section 1.2.2 within an extended scheme. This applies to both gradient-based and gradient-free approaches. In the former, consideration of a single scalar quantity to be minimized/maximized is usually essential, and thus any desire for the maximization and/or minimization of *multiple* objectives must be accomplished by mapping these into a single scalar value. This is most commonly achieved using the method of weighted sums, where the single scalar quantity for which derivatives are computed is taken to be the sum of each desired performance objective multiplied by some predetermined weight based on relative importance or other factors (Turevsky & Suresh 2011, Zhu, Zhang & Fatikow 2014). Variants of the weighted sums method known as compromise programming (Chen & Wu 1998) and physical programming (Lin, Luo & Tong 2010) have also been used for multiobjective topology optimization. However, the determination of suitable weights is nontrivial (Das & Dennis 1997, Messac, Sundararaj, Tappeta & Renaud 2000, Messac & Ismail-Yahaya 2001) and often requires a general intuition for the problem beforehand

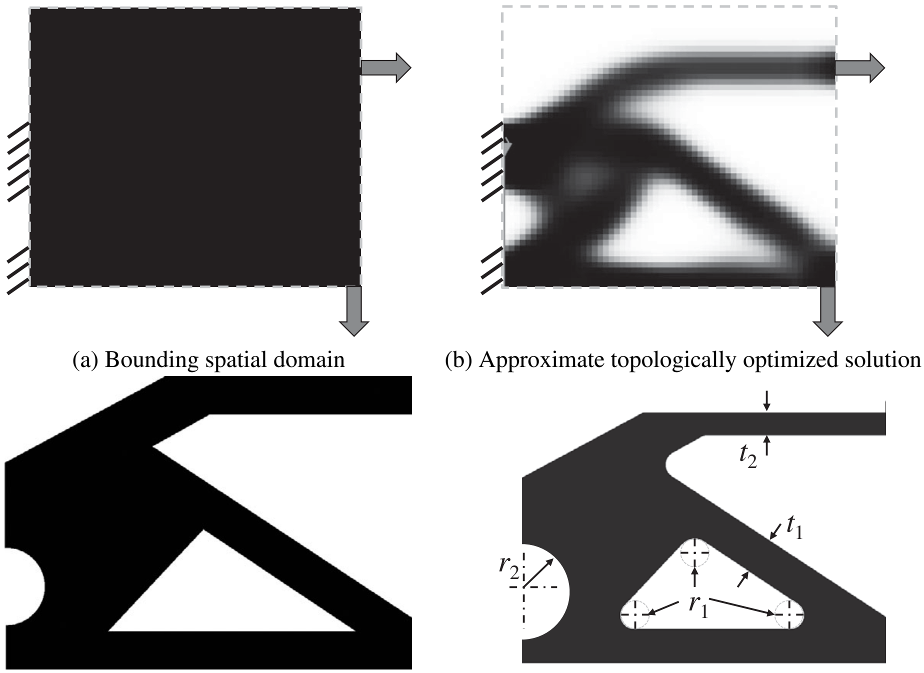
https://doi.org/10.1017/9781009026635.001 Published online by Cambridge University Press

(Tamaki, Kita & Kobayashi 1996). The single scalar value goal of gradient methods also dictates that a single optimal topology can be determined per optimization, such that each design along a desired Pareto frontier must be generated by an individual optimization process using a unique combination of objective weights. Furthermore, gradient methods using weighted objective functions have been shown to fail in capturing Pareto-optimal solutions for problems with non-convex Pareto frontiers (Das & Dennis 1997, Chen & Wu 1998, Messac et al. 2000, Messac & Ismail-Yahaya 2001).

As the name suggests, gradient-free optimization approaches do not require gradient information and instead solely use function evaluations of the objective function(s) to converge to a solution. One of the most common non-gradient methods, and the one most commonly utilized in topology optimization, is the GA, which is described in more detail in Section 1.3.3. GAs are motivated by the principles of natural selection and represent an optimization procedure that only requires a means of mapping design variable values to performance metric(s). (Tamaki et al. 1996, Deb 1999). This mathematical and algorithmic concept and its essential role in the overall bioinspired design approach captured in this work is introduced in Section 1.3. In short, GAs begin with a randomly generated set (or "population") of design solutions that are evaluated, assigned a fitness value based on their performance, and then modified by three operators borrowed from the fundamental ideas of genetics (selection, crossover, and mutation) to create a new and hopefully better population (Hare, Nutini & Tesfamariam 2013). This process continues until some specified termination criterion is met.

GAs are attractive when considering multiobjective topology optimization because they work with a population of solutions rather than a single solution, meaning that the set of solutions comprising the Pareto frontier can be obtained simultaneously. Their stochastic behavior also generally allows for these algorithms to better search the global design space, thereby avoiding convergence to local minima. Additionally, the lack of reliance on gradients makes GAs an appealing option when dealing with discrete design variables and discontinuous design spaces (Deb 1999), and GAs are perfectly suited to parallel processing, especially advantageous when using massive parallel architectures.

GAs are clearly essential to such previously listed algorithms as the Lamarckian genetic algorithm/SIMP hybrid method but have also been successfully coupled with other topology optimization approaches, including the level set (Guirguis, Hamza, Aly, Hegazi & Saitou 2015, Yoshimura, Shimoyama, Misaka & Obayashi 2016), ESO/BESO (Liu, Yi, Li & Shen 2008, Zuo, Xie & Huang 2009), and ground structure methods (Prasad & Diaz 2005). Sigmund has argued against their use in problems where the topology is represented explicitly by the design variables (e.g., SIMP, ESO/BESO) (Sigmund 2011), citing their computational cost relative to gradient-based methods and the necessity of using coarse meshes that cannot correctly represent the underlying physical response of the structure. However, as will be introduced in Chapter 2 and then demonstrated throughout the remainder of this work, they are essential to the use of developmental algorithms in topology optimization (see also Section 1.3).



(a) Bounding spatial domain

(b) Approximate topologically optimized solution

(c) Preliminary geometrically parameterized design

(d) Dimensionally optimized final part

Figure 1.4 Stages of the topological design process applied to a generic structural part. Note that the details of (b) strongly depend on the topology optimization algorithm selected (cf. Figure 1.3).

1.2.4 From Preliminary Topology to Detailed Design

Regardless of the algorithm chosen or whether one or more objectives is considered, topology optimization is becoming more commonly implemented at the preliminary stage of structural design (see Section 1.2.1). In most cases, topology optimization can determine optimal geometric configurations without well-defined initial conditions and/or a historical intuition as to the best solution. This is exemplified in Figure 1.4, which illustrates how the design process of a generic structural part can include a topological optimization step but also shows how such a step is not sufficient for full component definition (i.e., as for traditional manufacturing). Therefore, it is important to comment briefly on the relationship between topological optimization and full design.

Given a spatial design domain, a material system or system(s) to be considered, and a series of predefined boundary conditions that define the function of the component (Figure 1.4(a)), topology optimization can be used to determine an optimal distribution of material within the design domain that maximizes the performance of the structure (e.g., here, its stiffness) while satisfying the specified boundary conditions and not exceeding preexisting dimensional limits. The resulting "optimal"

https://doi.org/10.1017/9781009026635.001 Published online by Cambridge University Press

topological configuration (Figure 1.4(b)) can then be interpreted by designers experienced in parameterized design optimization and/or manufacturing limitations to generate a parametric geometric representation (Figure 1.4(c)). Additional studies (experimental, numerical, etc.) considering additional objectives, constraints, safety factors, etc. can then be performed to arrive at a finalized design (Figure 1.4(d)) having optimized thicknesses, fillet radii, and other dimensional features. In this way, employing topology optimization early in the preliminary design phase can lead to well-defined structures performing functions unique to a given design team.

1.3 Bioinspiration in Topological Design: The "EvoDevo" Approach

To this point, we have introduced the concepts of structural (or network) representation and topological design. Existing approaches for accomplishing the latter have been introduced, and the ideas of gradient-based, gradient-free, single-objective, and multiobjective optimization have all been addressed. The novelty and uniqueness of the content provided in this book is to propose generative approaches that hold promise for negating many of the disadvantages of these other methods, especially in the context of highly nonlinear, nonintuitive, multifunctional systems.

1.3.1 Motivation for Bioinspiration

Regardless of the tools utilized, the preliminary stage of structural design (cf. Figures 1.2 and 1.4) is largely influenced by designers' previous experience with or intuition for the problem at hand (Kaldate, Thurston, Emamipour & Rood 2006, Fantini 2007). A closer inspection of several of the methods described in Sections 1.2.2 and 1.2.3 reveals that topology optimization is no different. SIMP, level set, and GSM implementations all require an initial volumetric constraint, meaning that designers must have knowledge of approximately how much material they want to use. Each of the three methods is also typically reliant upon gradient-based optimization approaches, which require that for multiobjective problems the objectives be combined into a single scalar score using the method of weighted sums or similar. However, as previously mentioned, the determination of these weights is nontrivial and still requires that designers have a level of intuition regarding the desired outcome of the solution (Tamaki, Kita & Kobayashi 1996). SIMP and GSM implementations also employ direct subdomain specification (cf. Figure 1.1), meaning that the design variables directly define the topology of the structure. As such, the accuracy of the analysis and thus of the resulting topology is a function of the domain discretization, and both methods are plagued by rapidly increasing design space dimensionality and computation time when accurate solutions are required. For simple structural problems with a limited design space such as the example shown in Figure 1.4, such topology optimization methodologies may be sufficient. However, the maturation of adaptive material technology and a growing interest in developing structures or other networks with increased multifunctionality may begin to render these methods ineffective.

The goal of designing structures with multiple functionalities is inherently multiobjective, and the vast design space associated with these problems renders intuition formation extremely difficult. The use of traditional approaches (SIMP, GSM, etc.) would require a single objective to be formulated from multiple objectives (e.g., through the method of weighted sums), practically limiting designers to a small subset of all potential solutions. Furthermore, multifunctional systems typically respond to various stimuli such as stress, heat, electrical current/voltage, magnetic field, moisture, or light, and the complex physics models required to accurately analyze these responses greatly complicate the derivative calculations necessary for gradient-based topology optimization approaches. Therefore, there is a growing need for an inherently multiobjective preliminary design approach capable of exploring a vast design space to identify well-performing solutions to problems for which designers have little/no intuition or experience.

As the title of this monograph indicates, we propose bioinspired solutions to the various shortcomings of the currently established topological optimization approaches. The tools introduced and demonstrated herein, while varying in their details and utilities, all fall under a common framework that abstracts the design process into a biological analogy that is described in the following section.

1.3.2 The Evolutionary-Developmental Approach to Systems Design

In biological systems, the problem to be solved is "sufficient procreation toward propagation of a species," which requires survival (and even good health) of population members despite the presence of resource limitations, adverse environmental conditions, predators, intraspecies competition, and other challenges. As a means of formally introducing such bioinspiration into the design processes of interest, we here introduce the evolutionary-developmental or *EvoDevo* approach. As will be described, this analogy goes beyond simple "survival of the fittest" in terms of the selection, crossover, and mutation that comprise the common GA approach to optimization (introduced in Section 1.3.3). Rather, it also considers how genes, environment, heredity, and the laws of nature combine to drive the unique development of each new individual seeking survival. The full EvoDevo approach thus combines the developmental processes that govern spatial configuration (topological layout) with the evolutionary processes that improve populations over time, permitting great diversity in the context of straightforward algorithms. It is the *development* in particular that defines the unique topology optimization schemes described in Chapters 3–6 of this book.

As is appropriate for biological analogies to design, the EvoDevo approach is best suited for multiobjective optimization. In nature, individuals must perform a diversity of tasks associated with survival and procreation (e.g., hunting, courting, etc.) as they avoid critical pitfalls (disease, death, etc.). Some engineering designs must likewise perform multiple tasks without failure. In this work we will often focus on such *multifunctionality*, especially considering the optimization of networks, structural and otherwise, that seek multiple objectives and satisfy multiple constraints across

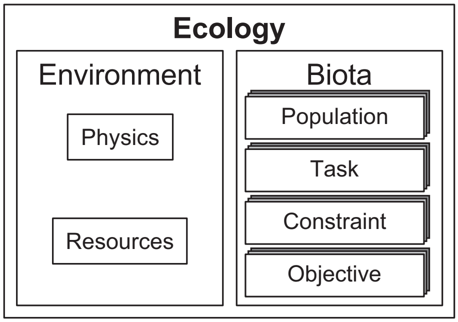


Figure 1.5 The EvoDevo abstraction considering both the living and nonliving aspects of the natural world, these concepts being interpreted for engineering design throughout this section.

multiple tasks.[5] This makes EvoDevo an effective alternative to traditional topological design approaches for multifunctional structures and systems.

The EvoDevo abstraction that captures the considerations of a multiobjective design problem at the highest level is depicted in Figure 1.5. Note that as we introduce these concepts and those that follow, we will use biological phrasing to represent engineering design concepts. At the top of the structure is the *Ecology* object, which is the cyber ecosystem within which design development and evolution occur. It represents the many choices of the designer in setting up a particular problem to be solved and, in this analogy, requires definition of two subobjects: the *Environment*, which represents the nonliving considerations within nature (physical laws and rigid constraints in engineering), and the *Biota*, which represents the relationships between living, and thus evolving, life-forms in nature (diversity of design options in engineering).

The *Environment* object abstracts the unchanging and impartial world that surrounds communities of living individuals. For us, it captures the designer specifications that apply to all processes and all individual designs. This object contains a *Physics* subobject, which includes the definition(s) of the physical phenomena to be considered, the mathematical models to be applied to these phenomena, the computational approaches employed in analyzing the models, and all physical constants. Also important are the *Resources* available to all designs and their assessment, including feasible ranges of all considered material properties, gross limitations on geometric parameters, and even computational resources (licenses, cores, time, etc.) available to the design problem.

On the other hand, the *Biota* object represents the relations of individuals to one another as populations of living entities evolve over time. While the *Environment* can be described as a negative directing force or constraint of evolutionary change, biotic

[5] To expand on the analogy, consider that a predatory bird's wing, for example, must enable both efficient soaring flight and high-speed controlled dives in adults, while permitting the compactness needed to fit inside an egg of minimum volume prior to hatching. Likewise, an airplane wing must generate lift at very low drag, carry these lifting loads without failing, and store fuel, among many other functions. In designing any engineering solution, multiple quantitative performance metrics can then be defined, many of them being contradictory with respect to designer's goals (e.g., a strong and robust wing that is also light).

competition helps populations to discover, propagate, and enhance the better individual designs over the poorer. Charles Darwin famously posited the predominance of biotic competition in organisms' "Struggle for Existence" to justify the idea of progress in nature (Gould 2002). In the *Biota* object, all evolving entities and the manner(s) in which some come to take preeminence over others (e.g., for the sake of final selection) are specified. Let us consider each subobject in terms of the guiding biological analogy and associated engineering design implication:

- *Population(s):* In nature, a population is defined as a collection of individuals of the same species; there may be many species (and thus many populations) in relationship with each other in a given ecology. In design, a population then represents a collection of individual designs associated with a given concept; there may be multiple design concepts (parameterized in a number of different ways) in the search for an optimal design.
- *Task(s):* Living organisms must complete a number of tasks to survive and procreate (e.g., eat, fight, court, travel, etc.), and their performance in these tasks relative to their peers and with respect to their environment determines their success. Engineered designs must likewise perform functions, where in this monograph we especially consider multifunctional designs as inspired by natural systems.
- *Constraint(s):* The performance of some tasks in nature must meet at least a minimum standard to allow survival and procreation. Animals must procure sufficient caloric intake and avoid contact with deadly predators, for example; these are limitations placed on them by their environment and biota. Most individual engineering designs are likewise developed to satisfy strict criteria, where failure to do so renders a design "infeasible."
- *Objective(s):* The success with which some tasks are completed by a living individual may increase or decrease the probability of procreation rather than allowing it or preventing it outright. Healthier (e.g., better nourished and less diseased) organisms produce more progeny, for example. In engineering design, an objective is a score by which individuals can be ranked and a "best" option is eventually selected for continued development. The consideration of multiple objectives, if applicable, explicitly precludes the down-selection of a single design but rather focuses the designer on a rational population of potential best choices (as during preliminary design).

Of course, the most complex entity in the entire ecology is the individual, a collection of which comprises a population. In nature, the unique definition of any individual is the result of both developmental and evolutionary processes; the information flow associated with these in the context of engineering topological design is illustrated in Figure 1.6. In traditional genetic design optimization of engineered systems, one will often formulate the modeled configuration of a given individual based *directly* on the design variables, neglecting the explicit consideration of developmental processes. As an engineering design text, this work likewise seeks to link design variables (genes) to expressed performance (fitness), but we intentionally propose a unique approach in which the *development* of the configurational layout as informed by genetic information is critically important. The inclusion of this developmental step is the

https://doi.org/10.1017/9781009026635.001 Published online by Cambridge University Press

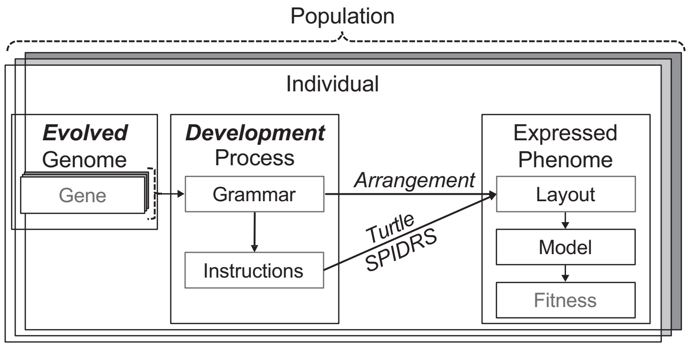


Figure 1.6 Population object and subobjects, specifically as interpreted in the context of engineering topology optimization.

essence of the *EvoDevo* approach and represents the primary bioinspiration driving this work.

As shown in Figure 1.6, the configurational definition of a design results from *Evolved* genetic information (*Evo*) as acted upon by a *Developmental* process (*Devo*). Together these result in the observable characteristics of the individual and, ultimately, the fitness thereof. Specifically, the *EvoDevo* approach taken throughout this book begins with the array of *Genes* (known as the *Genome*), which is determined through evolutionary processes and fully represents the uniqueness of a given individual. However, the genome alone is far from sufficient to fully define an individual and the full *EvoDevo* approach proceeds as follows:

1. Just as the position of every atom (or even every cell) in a natural organism is not defined by the genome but rather by the execution of chemical reactions and other physical processes based on genetic information, so too can information representing a unique engineering design be **translated** into a set of rules (i.e., a *grammar*) for developing a configurational layout. In the methods that follow, this will be achieved through the definition of the characters and rules of an *L-system* (see Chapter 3) or *arrangement L-system* (see Chapter 5).
2. Given these rules, the configurational (e.g., geometric) layout can then be accomplished in one of two **interpretation methods**. In nature, developmental processes (e.g., cellular division and differentiation) are directly informed by genetic information as they occur; in this way, the rules of development are iteratively applied directly to an iteratively changing configuration that, once complete, represents an individual. This direct grammar-to-layout approach in nature motivates the arrangement L-system topological representation described in Chapter 5. Alternatively, for reasons of algorithmic and conceptual simplicity, one can imagine an intermediate step during which gene-informed rules are fully interpreted into a long and highly complex set of layout-building instructions, which only upon completion are interpreted into a layout. This approach will be considered in Chapter 3 on Turtle graphics and Chapter 4 on the so-called SPIDRS algorithm, the former being quite

https://doi.org/10.1017/9781009026635.001 Published online by Cambridge University Press

straightforward to implement and the latter being much more capable as a network design tool.

3. The expressed phenome represents the observable characteristics of an individual (i.e., the body or physical extents of a living organism). In engineering design, it represents more than configurational *layout* information and a **modeling** process is required to assign other design-specific descriptors (e.g., localized material properties, section dimensional properties, the connectivity/connection properties at nodes, etc.) to the layout. A complete *Model* of an individual engineering design then represents the fullness of its unique essence as subjected to applied conditions associated with the Environment of the Ecology and the Task(s) (cf. Figure 1.5) for which performance is to be assessed. In this work, for example, an FEA model might be used to fully represent a design (i.e., its geometry and material composition) assessed against a function of interest (i.e., via applied far field and boundary conditions).
4. The final expression of an individual is its performance when assessed against Task(s) of interest. This **analysis** of the model provides the overall *fitness* relative to the Objectives and Constraints of the Ecology (cf. Figure 1.5).

To summarize in the context of engineering topological design, an array of genes known as a genome (the design variables) is **translated** into a grammar (development rules) that are then **interpreted** either directly or via instructions into a layout (geometric or network configuration). The layout is then **modeled** via the consideration of additional information, and the resulting model is **analyzed** under a set of imposed conditions to provide a final fitness (functional performance).

Given that the fitness of each individual is compared against the objectives and constraints of the Ecology such that better-performing individuals can be separated from poorer-performing alternatives, the evolutionary programming then acts on the population to evolve it between generations, as described in more detail below. Here we use crossover, selection, and mutation to discover, enhance, and propagate good features in the population. This is the ultimate goal of the design process. Relative to other more traditional approaches, it is the development process explicitly considered herein that is key for the effective search using a GA, as it is responsible for both the improved quality of design solutions and the accelerated convergence of the overall methodology (Pedro, Kobayashi, Coimbra, & da Silva 2008, Kobayashi, Pedro, Coimbra & da Silva 2009, Kobayashi, Pedro, Kolonay & Reich 2009, Kolonay & Kobayashi 2010, Pedro and Kobayashi 2011, Stanford, Beran & Kobayashi 2012). The fact that this bioinspired process is indeed effective for the design of a wide range of engineering networks (structural, fluidic, electrical, etc.) is the foundational thesis of this book and will be repeatedly demonstrated.

1.3.3 Genetic Algorithm Overview

Evolutionary algorithms in general are applied to a wide range of application areas from engineering, to art, to biology, to physics, and others. In the field of structural topology optimization, these algorithms have been used to improve the exploration

https://doi.org/10.1017/9781009026635.001 Published online by Cambridge University Press

of complex search spaces (Chapman, Saitou & Jakiela 1994, Kane & Schoenauer 1996, Schoenauer 1996, Jakiela, Chapman, Duda, Adewuya & Saitou 2000, Tai & Chee 2000, Azid, Kwan & Seetharamu 2002, Fanjoy & Crossley 2002, Hamda, Jouve, Lutton, Schoenauer & Sebag 2002, Tai, Cui & Ray 2002, Cappello & Mancuso 2003, Wang & Tai 2004, Madeira, Rodrigues & Pina 2005, Tai & Akhtar 2005, Wang & Tai 2005*a*, Wang & Tai 2005*b*, Wang, Tai & Wang 2006, Tai & Prasad 2007, Balamurugan, Ramakrishnan & Singh 2008, Guan & Chun 2009, Zuo et al. 2009, Madeira, Pina & Rodrigues 2010).

Relative to the previous discussion, evolutionary optimization algorithms are biological metaphors that can produce high quality designs by identifying, recombining, and enhancing the best features present in a continuously adapting population of individual designs. Throughout this book, we will only make use of GAs , taken to be those heuristic approaches that specifically apply the processes of selection, crossover, and mutation to sequential populations of designs such that new, and hopefully improved, generations continue to be generated. The execution of such an algorithm encompasses all aspects of Figures 1.5 and 1.6. In the context of the GAs employed in this monograph, evolution starts with a population of individuals, often generated via a random or quasi-random manner (e.g., via Latin hypersquare sampling (Liefvendahl & Stocki 2006)), and then proceeds with the following steps:

1. Each member of a full population, being uniquely described by its individual genotype, is subject to the four-step translation–interpretation–modeling–analysis process described above such that all fitnesses can be quantified. In the context of engineering, this will involve the calculation of one or more performance metrics based on the physical response in question (e.g., structural, fluidic, electromagnetic, thermal).
2. The relative fitnesses of all members of a population are then compared, and a *selection* algorithm ranks the individuals, determining the best performers based on some predetermined criteria that may consider both response preferences and diversity preservation.
3. Selected designs are then assigned to pairs and their genotypic content combined and used to create one or more offspring based on a number of potential *crossover* algorithms.
4. Finally, the genotypes of the new offspring are most often subject to *mutation* to further increase design diversity. Sufficient crossover and mutation operations will eventually lead to a new population of desired size.

The cyclic evolutionary process continues when this new population is likewise evaluated to quantify its relative fitnesses. As this process emulates natural selection, populations tend to improve in the relative performance of their individuals until designs converge toward optimal configurations.

For the purpose of mathematically defining the kind of multiobjective optimization problem that will be solved throughout this text using the EvoDevo approach, we formally introduce the following four classes of variables:

- **Design Variables:** These are the components of $\boldsymbol{x}$ that represent all genotypic content subject to definition by a designer. In the methods on which this overall text is based, these will be shown in the following chapters to be either (i) aspects of the grammar (i.e., L-system) to be defined in the following chapters, (ii) graphing algorithm parameters, or (iii) physical parameters of the system to be modeled, especially as related to interpretation of genotype into phenotype (i.e., the modeling step of Figure 1.6).
- **Bounds:** These are the upper limits $\boldsymbol{x}^{ub}$ and lower limits $\boldsymbol{x}^{lb}$ on $\boldsymbol{x}$ such that $x_i^{lb} \leq x_i \leq x_i^{ub}\ \forall\ i$. In the current explorations, these will be defined in a straightforward manner relative to the L-system assumed or guided by *a priori* understandings regarding physical parameters (cf. *Resources* in Figure 1.5).
- **Objectives:** These components of the set $\boldsymbol{f}$ are the performance metrics with respect to which the designer has a preference (i.e., either minimize or maximize) but for which no strict requirements apply. Given an unchanging physical model considering an unchanging set of operational conditions, the objectives are fully defined by the selection of $\boldsymbol{x}$ such that one may express them as $\boldsymbol{f}(\boldsymbol{x})$ (cf. Figure 1.5).
- **Inequality Constraints:** These components of $\boldsymbol{g}$ quantify the satisfaction of strict design requirements by exploiting the fact that each requirement considered can be strictly expressed in the form $g_i \leq 0$. For example, if the maximum Mises stress $\bar{\sigma}$ calculated within a structure must be held to below a known failure stress σ_f, then the ith inequality constraint could be written as $g_i = \sigma_f - \bar{\sigma} \leq 0$. As with the objectives, these constraints (cf. Figure 1.5) may be expressed as $\boldsymbol{g}(\boldsymbol{x})$.

Note that a set of *equality constraints* (i.e., $\boldsymbol{h}_i = \boldsymbol{0}$) can also be considered but are rarely of utility in the design of engineered structures except as may be included implicitly in the calculation of $\boldsymbol{f}$ or $\boldsymbol{g}$.

With these variables defined, the multiobjective optimization problem can then be stated as follows:[6]

$$\begin{aligned} & \boldsymbol{f}^* = \min_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}^*), \\ \text{such that} \quad & \boldsymbol{x}^{lb} \leq \boldsymbol{x} \leq \boldsymbol{x}^{ub}, \\ & \boldsymbol{g}(\boldsymbol{x}) \leq \boldsymbol{0}. \end{aligned} \tag{1.1}$$

A design $\boldsymbol{f}^* = \boldsymbol{f}(\boldsymbol{x}^*)$ is said to satisfy Equation 1.1 if there is *no* $\boldsymbol{x} \neq \boldsymbol{x}^*$ such that

$$f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{x}^*)\ \forall\ i.$$

Such a point cast into the space spanned by $\boldsymbol{f}$ is known as "nondominated." In the idealized case of full convergence, the set of all such nondominated points is known as the "Pareto frontier" (see, for instance, Coello 2002).

The evolution of sequential populations of 10 designs from an initial randomized guess toward a converged Pareto frontier is schematically shown in Figure 1.7. With

[6] Here we assume preference for the *minimization* of all objectives. This is without the loss of generality as any preference for maximization (e.g., maximum compliant displacement or maximum strength) can be mathematically converted to a preference for minimization via multiplication of such an objective by (-1).

https://doi.org/10.1017/9781009026635.001 Published online by Cambridge University Press

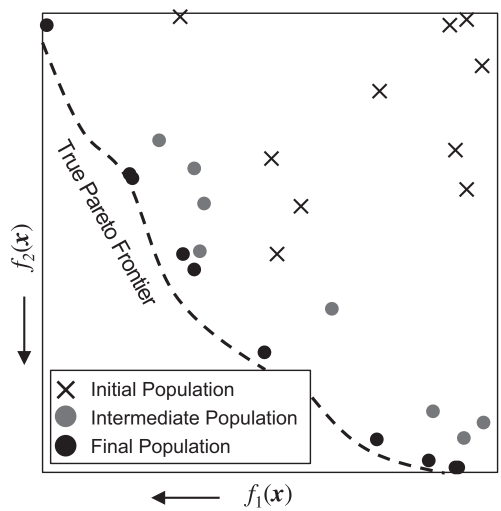


Figure 1.7 Schematic illustration of the evolution of a population of designs toward simultaneous minimization of two competing objectives f_1 and f_2. The idealized converged set of all such points accomplishing this goal (i.e., satisfying Equation 1.1) is known as a “Pareto frontier,” the performance of which may not be reached by even the most evolved designs (black points).

reference to the general process for a GA previously described, designs that are on or near the current approximation of the Pareto frontier and that do not violate constraints (not shown) are retained for crossover, leading to an ever-improving population. Note that both simultaneous minimization of the components of $\boldsymbol{f}(\boldsymbol{x})$ and an even distribution of points along the resulting frontier (i.e., the maintenance of diversity) are both goals of many multiobjective GAs, including the widely used NSGA-II algorithm (Deb 2001). In general, as the true Pareto frontier is unknown, a genetic optimization process will terminate before its final population performs at this highest possible level, as shown in Figure 1.7.

1.4 Summary of This Book

The remainder of this book seeks to introduce and describe how various algorithmic schemes can be used to implement an EvoDevo approach for the design of multifunctional structures and other networks with the advantages of compact representation and complete general applicability across system physics and number of functions or objectives considered.

- To provide a comprehensive introduction to existing topological optimization approaches, **Chapter 2** first demonstrates the formulations and features of many of the options listed in Figure 1.3 in addition to others;

https://doi.org/10.1017/9781009026635.001 Published online by Cambridge University Press

- The concepts of L-systems, straightforward graphical interpretation, and GA-based optimization over such a framework are introduced in **Chapter 3**;
- **Chapter 4** then shows how L-systems and optimization can be applied to produce much more effective networks in two and three dimensions by developing a novel graphical interpretation scheme (SPIDRS);
- A distinctive approach whereby execution of the rewriting process is manifested directly as graphical development rather than as character string modification (the arrangement L-system, cf. Figure 1.6) is then presented in **Chapter 5**;
- Finally, **Chapter 6** demonstrates the use of the various bioinspired algorithms presented in solving a range of multiobjective multiphysical problems.

Most importantly, it will be shown that the most widely accepted methods of topological optimization (i.e., SIMP and level set) can be outperformed by bioinspired approaches for a range of structural design problems, especially when low area or volume fraction designs are most suitable or multiple physical (and especially nonstructural) responses are of interest. It will also be shown that classical sensitivity approaches, such as the GSM can also leverage bioinspired approaches as the latter are capable of generating high-performing but nonintuitive ground structure configurations.