Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2015), *29*, 393–415.
© Cambridge University Press 2015 0890-0604/15
doi:10.1017/S0890060415000402

Evolutionary structural and spatial adaptation of topologically differentiated tensile systems in architectural design

SEAN AHLQUIST,[1,2] DILLON ERB,[1] AND ACHIM MENGES[2]

[1]Taubman College of Architecture and Urban Planning, University of Michigan, Ann Arbor, Michigan, USA
[2]Institute for Computational Design, Stuttgart University, Stuttgart, Germany

(Received October 1, 2014; Accepted July 13, 2015)

Abstract

This paper presents research in the development of heuristic evolutionary algorithms (EAs) for generating and exploring differentiated force-based structures. The algorithm is weighted toward design exploration of topological differentiation while including specific structural and material constraints. An embryological EA model is employed to "grow" networks of mass-spring elements achieving desired mesh densities that resolve themselves in tensile force (form-active) equilibrium. The primal quadrilateral quadrisection method serves as the foundation for a range of extensible subdivision methods. Unique to this research, the quad is addressed as a "cell" rather than a topological or geometric construct, allowing for the contents of the cell to vary in number of mass-spring elements and orientation. In this research, this approach has been termed the quadrilateral quadrisection with *n* variable topological transformation method. This research culminates with the introduction of a method for *grafting* meshes where emergent features from the evolved meshes can be transposed and replicated in an explicit yet informed manner. The EA and grafting methods function within a Java-based software called *springFORM*, developed in previous research, which utilizes a mass-spring based library for solving force equilibrium and allows for both active (manual) and algorithmic topology manipulation. In application to a specific complex tensile mesh, the design framework, which combines the generative EA and mesh grafting method, is shown to produce emergent and highly differentiated topological arrangements that negotiate the specific relationships among a desired maximal mesh density, geometric patterning, and equalized force distribution.

Keywords: Evolutionary Algorithm; Force Networks; Form-Finding; Subdivision; Topological Differentiation

1. INTRODUCTION

The research described in this paper presents a computational method for designing topological variation in prestressed structural meshes through the use of evolutionary algorithms (EAs), mass-spring based simulation and a Java-based modeling environment called *springFORM*. This research is motivated by an interest in developing design-oriented computational tools for architectural systems where form is driven by the interaction of structure and materiality. The research focuses on the ability to design surface structures formed as pure tensile meshes. Inherent in this research is the inextricable relationship among the form of the surface, the number of elements that comprise the mesh definition of the surface, and the movement of tension forces through the mesh. The reciprocity among these three factors introduces the primary challenge for this research: any shift in the mesh configuration or alteration in the number of mesh elements drives a change in the organization, geometry, and structure of the surface. Therefore, the critical problem being addressed is the ability to make informed design decisions regarding mesh topology and its relationship to form, specifically for cases using nonperiodic meshes with a high number of elements. The ability to decipher the ramifications of topology manipulation to structural performance and geometric form becomes overwhelming as the number of elements and the irregularity in the mesh increases. This research proposes the use of an EA in combination with a manual *grafting* tool, where an informed design process emerges through the ability to assemble evolved nonperiodic mesh patches into tensile surfaces.

The software program springFORM was developed in previous research to enable active manipulation of spring-based meshes during the process of "form-finding" in order to allow

Reprint requests to: Sean Ahlquist, Taubman College of Architecture and Urban Planning, University of Michigan, 2000 Bonisteel Boulevard, Ann Arbor, MI 48109, USA. E-mail: ahlquist@umich.edu

https://doi.org/10.1017/S0890060415000402 Published online by Cambridge University Press

for topology, form, and structural relationships to be visualized, understood, and ultimately codified. Various periodic topologies and levels of subdivision can be implemented algorithmically, allowing for a range of mesh densities to be explored. However, these functions operate only on the global scale of the entire mesh. The type of mesh topology and the level of subdivision are constant within a single mesh. The periodic topology can be altered manually on a local scale, providing the ability on a step-by-step basis to view how individual changes in mesh organization shift the force distribution and produce a new form. Yet, each change in the mesh influences the effectiveness of previous changes, possibly nullifying them. Therefore, at each alteration, all previous alterations need to be reassessed and potentially realtered. It is this cyclical nature, while considering simultaneously how topology affects form and structure, that makes such an approach ineffective as a design-oriented semi-algorithmic tool for dealing with differentiated tensile meshes.

Mesh density is a critical design factor in determining appropriate mesh topologies. Maximal densities are desired in order to allow for a porous mesh to visually describe a continuous surface structure. The porosity of a mesh simultaneously produces a transformative visual description, where layers of the doubly curved surfaces collapse and separate based on the angle of view (Fig. 1). Constructability is considered in how and at what intensities the tensile forces move across the mesh surfaces. The desire is to minimize the difference between maximal stresses that arise at the anchor points and minimal stresses that exist at the most interior portions of the tensile meshes. In minimizing this differential, an entire mesh can be constructed of a single material type. In practice, measuring the linear range of elasticity for a single cable material defines the minimum and maximum limits of allowable stress in the structure. Constructability and density are interrelated. In previous work, density was defined by the smallest length of material possible between nodes. Within the given periodic mesh and level of subdivision, peak stress where allowable was based on the properties of the cable. In this instance, meshes have been generated using a periodic "dia-grid," where the determination of density is based upon a global subdivision variable, where the same nodal condition of always connecting four mesh edges at a single node is maintained. In the research described in this paper, the question of how to computationally produce and manage meshes with n number of mesh edges at a single node is addressed. The intent is not to directly address overall mesh density, rather to evolve the proper placement and topological construction of differentiated mesh densities. In exploring and evolving efficiencies in structural frames (vector-active trusses) and meshes (form-active cable meshes), it is critical to test varieties of regular and irregular topologies (von Buelow, 2008). In designed-oriented cases where mesh topology is the critical feature for optimization, exploration is often accomplished by testing topological variation based on a single mathematical, periodic method (Schein & Tessman, 2008). This research seeks to develop a fundamental approach, unconstrained by a single mathematical logic, whereby a wide variety of nonperiodic patterns can be realized within a single topology.

SpringFORM is positioned to allow for tacit knowledge of mesh structure and structural form to be developed through active visualization and feedback among topology manipulation, tensile forces, and resulting form. What can be termed as a method based on *embedded rationality*, the means of form generation and the measure of design effectiveness are col-

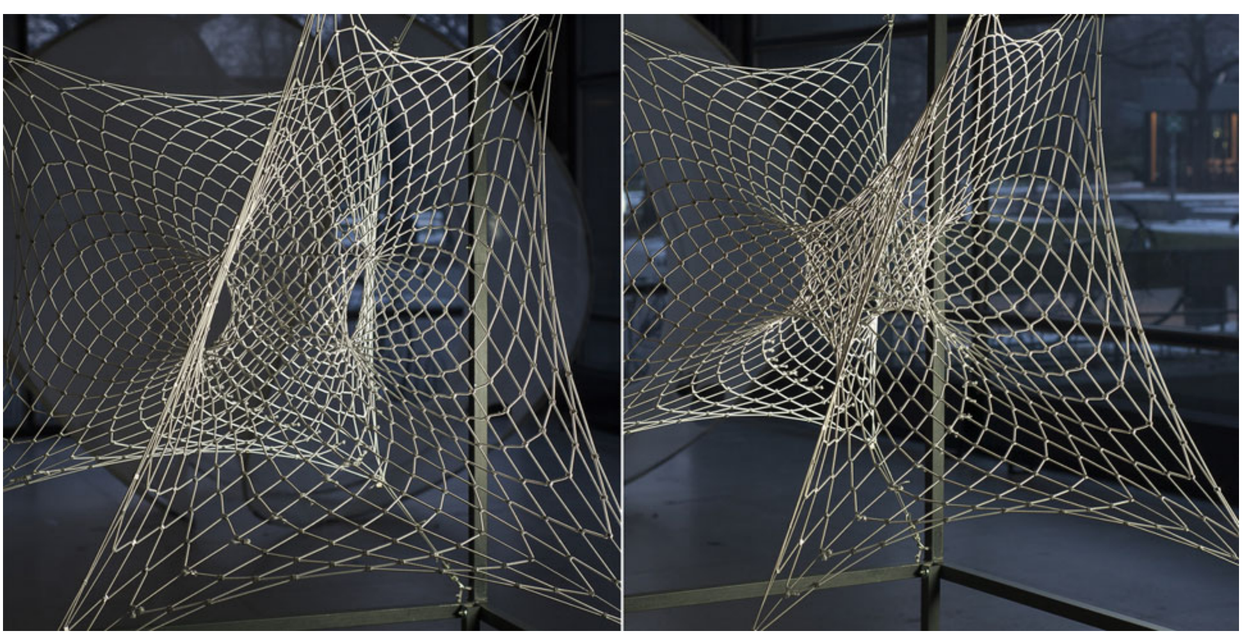

Fig. 1. Mesh density is a desired trait for the tensile meshes developed in this research, allowing for surface continuity to be readable, as well as a transformative nature of overlapping layers from different viewing angles, accomplished here with a periodic "dia-grid" mesh. (Cylindrical mesh morphology prototype, Sean Ahlquist, Institute for Computational Design, Achim Menges, University of Stuttgart, 2009).

https://doi.org/10.1017/S0890060415000402 Published online by Cambridge University Press

lapsed into a single iterative process (Aish et al., 2012). In addressing the possibility for nonperiodic meshes, learning through iterative topology manipulation is limited as described above. Where form is realized at the complex interaction of topology, materiality, and structural behavior, certain potentials can only be discovered through exhaustive means of iterating through variations across each of these form-critical parameters. Therefore, the EA and *grafting* methods described in this paper offer a bridge for exploring immensely vast design spaces, providing means where tacit learning can still be invested within a largely algorithmic process. However, a key factor in this research is to not consider the EA as the sole, autonomous design generator. Rather, the EA serves to generate meshes of a maximal density with certain structural efficiencies, where in a subsequent design step, patches of meshes can be supplanted from the evolved mesh into other meshes. This *grafting* tool is pivotal in allowing for *manual* design manipulation to occur at a mesoscale, operating with groupings of evolved springs rather than individual springs or the entire mesh topology.

1.1. Integrating modes of physical and computational design

In designing form at the relationship of material properties and structural behavior, this research is built upon deploying several modes of design, spanning study from physical models to computational simulation. Experimentation with physical models serves as the primary means to establish tacit knowledge and initial precedents for system-specific parameters such as material choice, structural strategy, and even geometric relationships. These particular parameters can be categorized as topology, structural action, and materiality (Ahlquist et al., 2013; Ahlquist & Menges, 2013). Within a computational environment, deviations from the physical precedent can be extrapolated by establishing ranges of variables for topology, structural action, and materiality from the physical exemplar itself. In application to the study of prestressed structural systems, physical form-finding serves to develop initial design precedents for which further exploration takes places in mass-spring based simulation and finite element analysis (FEA). To construct a design framework, it is necessary to understand the constraints of preplanning and topology related to each mode of form-finding. *Preplanning* is defined as the consideration of how much geometric information is necessary at the outset in order to form-find a solution. *Topology* defines the number of elements in the system and their association to one another, regardless of dimension or position. By examining the level of topological complexity in the design and comparing it to the degree of preplanning necessary for each mode of design, the level of engagement and sequencing among physical, spring-based, and FEA form-finding can be determined (Ahlquist et al., 2014). Depending on the mode of design and material implications, certain elements may require geometric definition in advance of implementing structural action and ultimately generating form. Conventionally speaking, FEA requires the most preplanning in the setup of both topological and geometric data, whereas physical form-finding and mass-spring simulations allow more freedoms in active manipulation of such parameters during the form-finding process (Fig. 2). For the structural studies in this research, the preplanning aspect is intensive for an FEA model, where the full geometric description produced by the spring model is needed. Its positioning within this design process would be as a step to validate results rather than to iteratively explore design solutions. Ultimately, each mode of physical and computational design contributes a certain aspect of a *composite learning curve*, shifting between manual and automated algorithmic procedures (Aish, 2011).

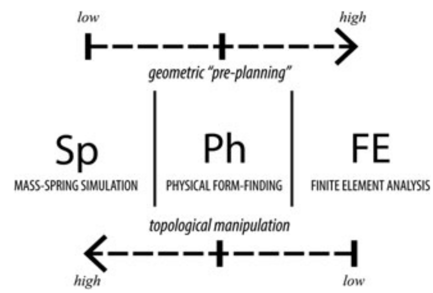

Fig. 2. For cases of topological complexity, geometric "preplanning" increases while the ability to actively manipulate topology decreases when shifting between modes of design from mass-spring simulation to physical form-finding and finite element analysis.

2. BACKGROUND

2.1. SpringFORM and mass-spring based simulation of complex topologies

SpringFORM is a software program written in Processing (Java). It was developed to allow for complex mesh topologies to be readily generated, manipulated, and activated as pure-tensile meshes (Fig. 3). The simulation of tensile and other force-active behaviors is accomplished through the use of a preexisting library developed by Simon Greenwold (2009), which uses a particle (or mass-spring) system based upon Hooke's law of elasticity to compute forces by measuring the degree of displacement from a given rest length (Terzopoulos et al., 1987). The method is skewed toward stability in calculation, producing approximated results (Kilian & Oschendorf, 2005). Complex mesh topologies are generated in springFORM as networks of springs, where various structural behaviors such as prestressed tensile (form-active) and bending-active structures are computed with the particle system library (Ahlquist et al., 2013; Ahlquist & Menges, 2010). SpringFORM provides a layer on top of the particle system library to organize complex meshes into objects and provide an interface for actively manipulating the topologies and spring-based properties at the particle level, the spring level, or higher order collections of springs and particles such as cylindrical meshes. It is of the most importance that this is de-

https://doi.org/10.1017/S0890060415000402 Published online by Cambridge University Press

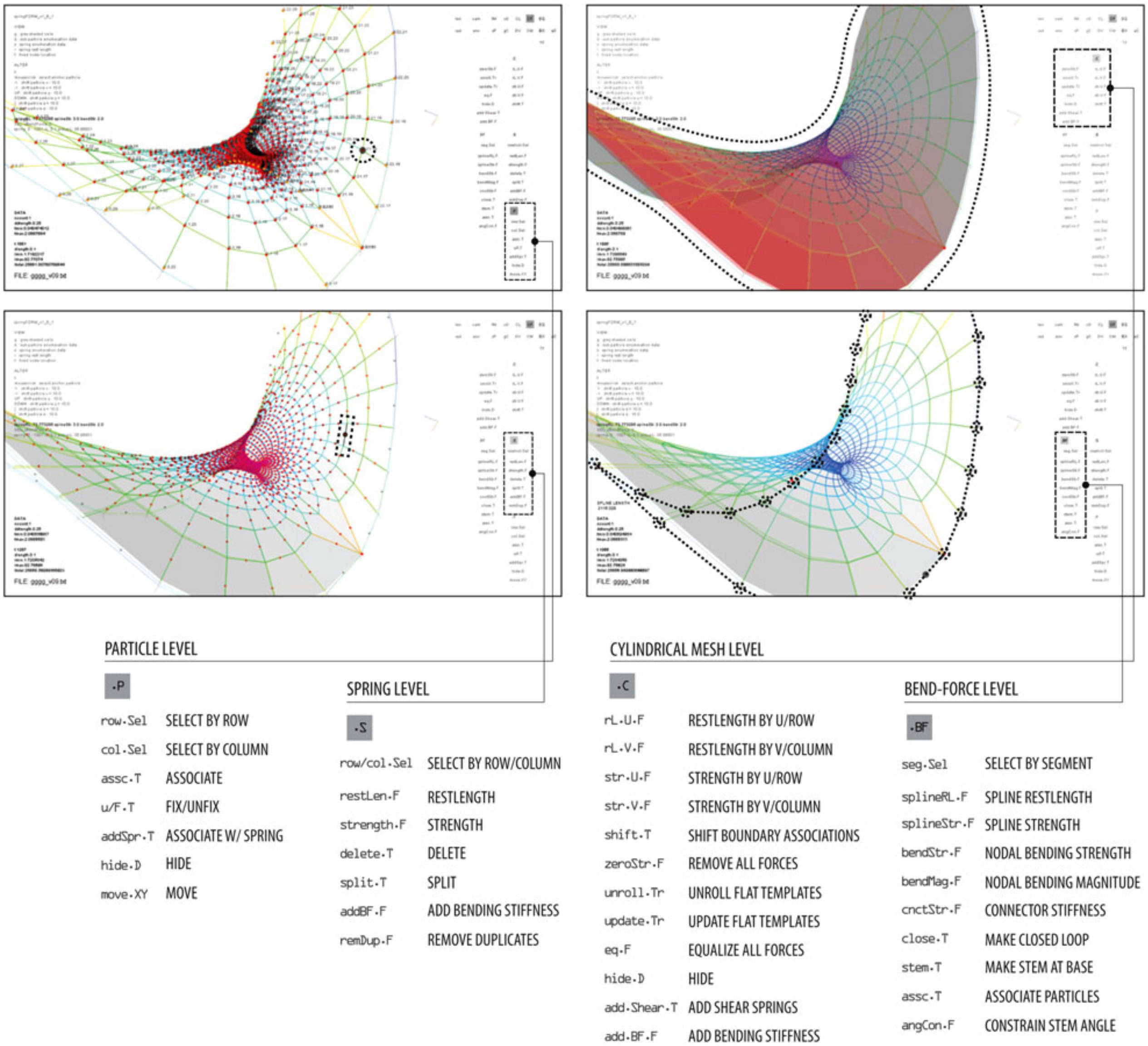

Fig. 3. SpringFORM is software written in Processing (Java), which allows for complex meshes to be generated and manipulated as networks of forces, primarily simulating behaviors of tension and bending stiffness, utilizing a particle system library developed by Simon Greenwold to compute the interaction of Hookean forces.

signed so that manipulation may occur *during* the process of form-finding. Via manual manipulations of the spring model, incremental adjustments are immediately visualized for their repercussions in form and structural behavior. Such immediate feedback helps to dissolve the complexity in comprehending how changes in force distribution simultaneously shift structural behavior and form. This is unique in comparison to other computational methods, where often both geometric and topological descriptions must be fixed before calculation can occur. Very little geometric preplanning is necessary because of the springs' abilities to have linear (and infinite) elasticity, and the tools within the springFORM environment for actively manipulating topology, in the manner by which springs are interconnected. Therefore, in the case of tensile meshes, a solution of force equilibrium will always be found, though it will not be constrained by explicit material properties.[1] In this sense, springFORM provides a more fluid bridge between methods of manual form-finding with physical models to more precise calculations with engineering-oriented software, such as FEA. Within a design process, it is situated as a mode for design exploration of advanced topological complexity and variation based upon initial precedents from physical prototypes. The value of the mass-spring engine and its deployment within the springFORM modeling

[1] The particle and mass-spring simulation within springFORM relies on solving a linear system with an "implicit Euler" iterative method (Baraff & Witkin, 1998). This allows the system to maintain stability even with very large deformations. The actual implementation relies on the biconjugate gradient stabilized iterative method for solving the linear system. The springFORM software utilizes the SimonG particle system library (Greenwold, 2009).

https://doi.org/10.1017/S0890060415000402 Published online by Cambridge University Press

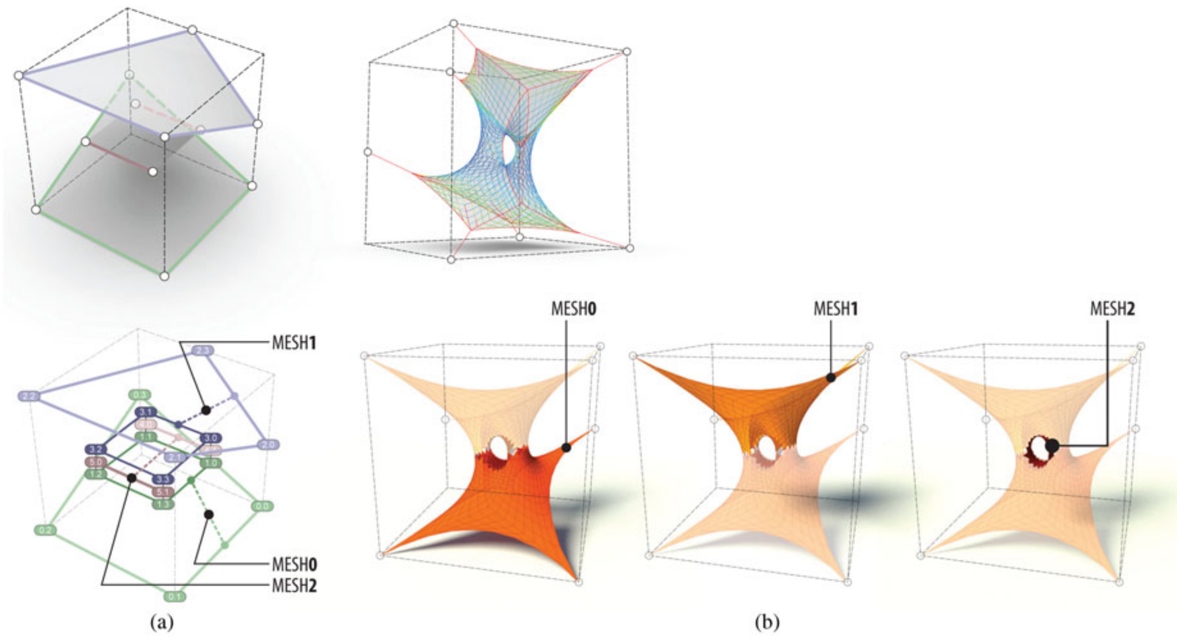

Fig. 4. (a) Metatopology "framework" to describe boundary relationships between cylindrical meshes (exploded framework below), with (b) the resulting spring mesh, in the springFORM software program, based on the local-topology of a dia-grid and built of three interconnected cylindrical meshes.

environment is to enable topological complexity to be explored beyond the limitations found in physical form-finding, where constructing iterative design models is an exhaustive process, and FEA, where multiple key factors need to be known and resolved prior to initiating the form-finding process. As the studies show, physical form-finding can suffice to explore and resolve complexity of multiple interconnected elements loaded under tensile forces. The challenge, though, is to augment such explorations by studying variations and expansions of topological complexity beyond the initial studies. Therefore, in this research, the starting point, of involving a heuristic algorithm, is at a level of topological complexity that goes beyond that which is readily attainable through physical form-finding.

To test the methods developed in this research, an intricate tensile mesh structure is utilized as the primary test case, exhibiting a level of complexity that would be difficult to initially generate as a physical model. The initial design for the prototype is generated through the springFORM software as a complex *metatopology* of multiple interconnected cylindrical meshes. Each individual mesh has a *local topology* based on a regular *dia-grid* mesh (Fig. 4). The prototype, which is called cylindrical mesh morphology, depicts a spatial intention of controlling the size and directionality of the cylindrical apertures, as well as a quality of mesh continuity and material density. This research intends to provide means by which such integrated qualities of form can be maintained, if not further explored, while structural considerations of material robustness are addressed.

Embedded in the organization of the topology for the cylindrical mesh morphology prototype is the logic for materializing the structure as an elastic cable mesh. In analyzing the structural properties of a 2.5-mm diameter polyester/elastan cord, the forces defined in the spring mesh are translated into material lengths. There is an explicit translation where each spring in the springFORM model is defined as an individual length of material. The cylindrical diagrid topologies are unwrapped into individual rows and mapped to show both local nodal associations that define the dia-grid and global associations that define topological relationships at the boundaries of the cylinders (Fig. 5). By equating the amount of material stretch to a value of stress in Newtons per square millimeter (N/mm^2), the information for force in each element in the computational model can be translated to a static (unstressed) material length of the elastic cable. Although constructed with a regular topology, the resulting form comprises highly differentiated stresses, thus differentiated material lengths. The materialization process in constructing the physical prototype exhibits the intense degree of preplanning necessary to produce a well-resolved form in tension equilibrium (Fig. 6). This exemplifies the gauge provided in Figure 2 where the mass-spring based mode of design serves as the ideal avenue for exploration due to the minimal amount of geometric preplanning necessary combined with the maximum amount of topological manipulation available, for cases where high degrees of topological complexity are being explored.

2.2. Topological variation in force networks

When topology is considered as a variable in the number of springs that can connect at each individual node, this results

https://doi.org/10.1017/S0890060415000402 Published online by Cambridge University Press

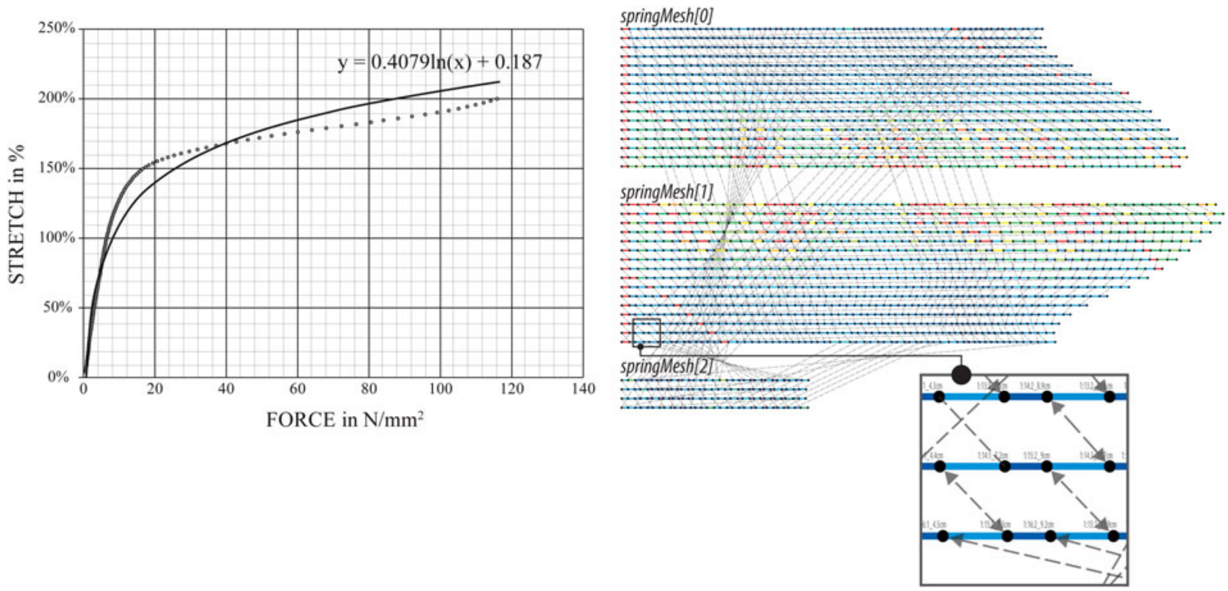

Fig. 5. (Left) Digital testing of force to displacement of 2.5-mm diameter to elastic cord, extrapolated to (right) the layout of mesh topology indicating material lengths and nodal associations.

is an intensely expansive design space. This is magnified by the desire to achieve dense concentrations of mesh, which is the level of complexity at which this research intends to operate. Where each element has a material and explicit structural influence, variation in connectivity simultaneously shifts structural behavior and the resulting form, as shown by the changes in form and performance between a quad-grid and a dia-grid mesh for a multicylinder metatopology (Fig. 7). Because the mesh is a continuous network, force is also continuous, meaning a change in one location of the mesh structure will produce ramifications throughout the entire system. Therefore, topology cannot be managed only locally; changes must always be addressed with regard to their ramifications to the entire system. Such manipulation is essential, though, where regular mesh topologies are ill suited to resolve peak stresses where forces accumulate extensively at individual nodes. To accomplish the prototype in Figures 1 and 4, the peak stresses were clamped to enable the entire system to be made of a single material, resulting in noticeable geometric discrepancies at the border and anchor locations. The ac-

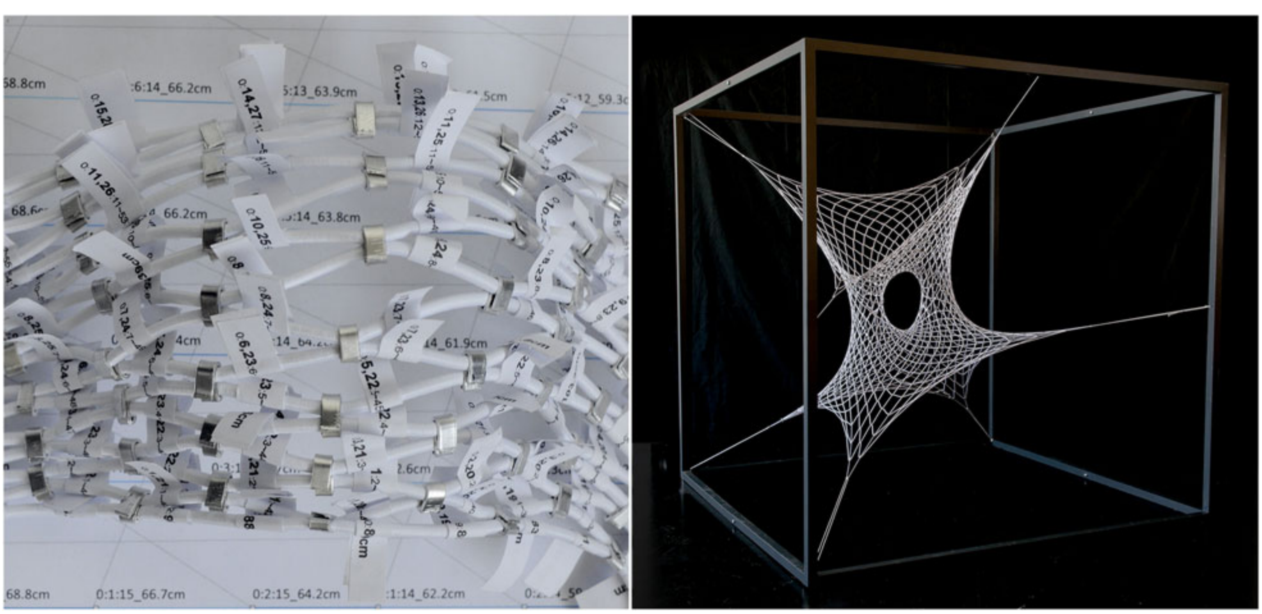

Fig. 6. (Left) Reconstruction of cable mesh from mapping and geometric data extracted from springFORM model; (right) tensioned into complex multicylindrical form.

https://doi.org/10.1017/S0890060415000402 Published online by Cambridge University Press

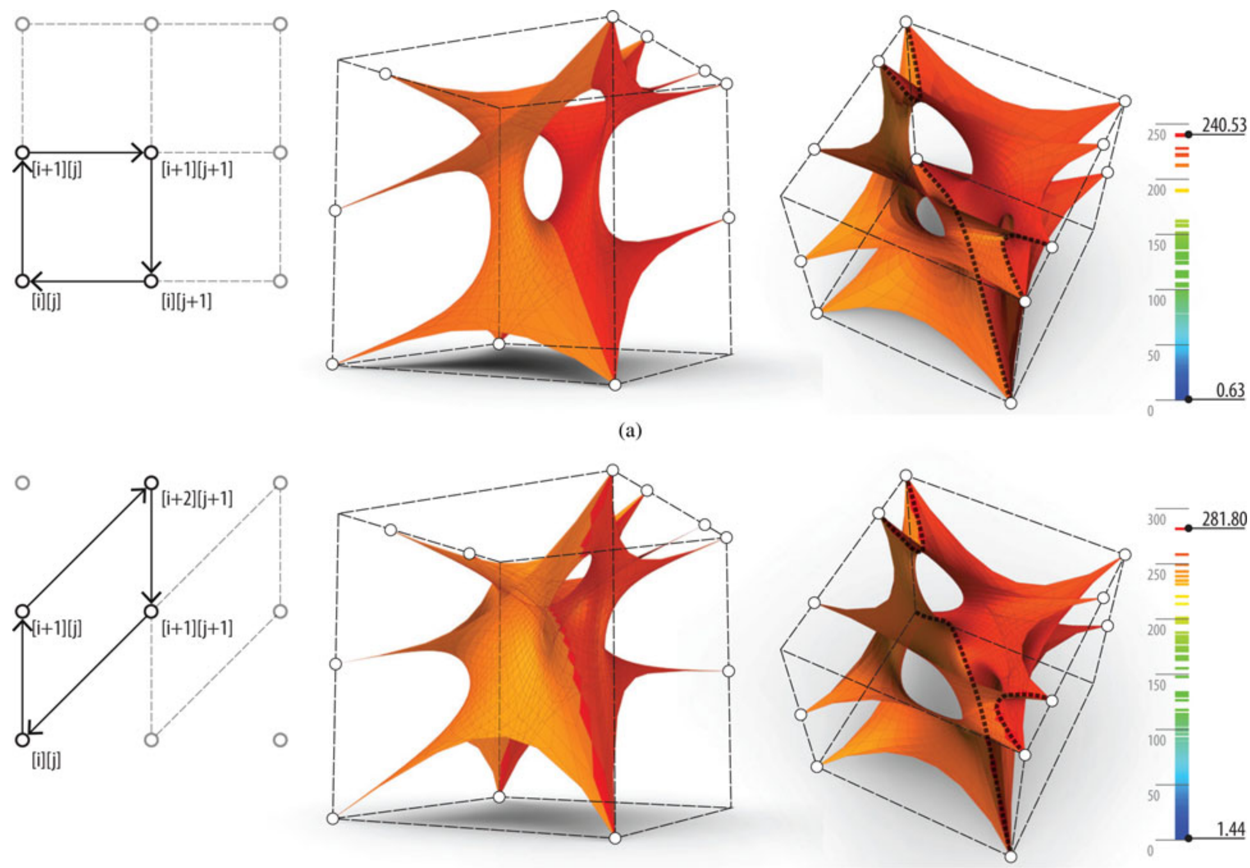

Fig. 7. Comparison of tensile meshes based upon the same metatopology, only with variation in the local topology between (a) a regular quad-grid logic, and (b) a regular dia-grid logic, which leads to a difference of almost 200% in relative force differential, driven by concentrations of areas with low stress and boundary springs with high stresses.

tual force differential in the spring model showed the peak stress to be 90 times bigger than the most minimal stress in the mesh structure, with the examples below showing even more significant force differentials. This critical aspect of performance is explored in this research, addressing both peak stresses and concentrations of springs with minute force exertion that equally contribute to values of significant force differential. The basis for effective means is determined by resolving such force discrepancies with the use of highly irregular spring-mesh topologies.

The combined effort of introducing mesh density and manipulating topology introduces a unique challenge in searching for effective solutions. In this research, a subdivision scheme is introduced, allowing for mesh resolution and topological variation to be introduced in a single step. In a series of manual operations of applying subdivision at anchor points, a dramatic and undesirable geometric change of bundling occurs. The forces bundle into geometrically narrowed pathways in an attempt to mediate the singularity of stress at the anchor point with the distributed forces within the larger mesh. In stepping around the multiple anchor points of a mesh, the overall performance of force differential is shown to dramatically worsen as each anchor point is addressed in succession. Even though the stresses are handled in a more equalized manner at each anchor point, by introducing new springs, the overall force in the system is increased and the other anchor points exhibit more significant stresses. Only until all anchor point conditions have been resolved does the overall performance improve. This poses the unique challenge where density is desired yet it shrinks surface area, and resolving stress means the system has to go through steps of getting worse until it finally starts to exhibit overall improvement (Fig. 8). These simple topologies exhibit clear and observable impact among topology, force, and resulting form. Yet through only these simple topological operations, a search space of possible solutions is not so large that an optimization algorithm is necessary, where it could most likely be traversed through only a brute-force method. By contrast, the multicylinder example shown in Figure 7 is a collection of three separate cylinder topologies, where it is difficult to ascertain the manner in which topological differences are altering form and structural behavior. Furthering this discussion into cases where irregular topologies are used, forces (as defined by the topology of springs) would not organize along any clear axes, such as the UV coordinate system of the simple cylinder, and much more topologically diverse design space is produced. It is at this moment that

https://doi.org/10.1017/S0890060415000402 Published online by Cambridge University Press

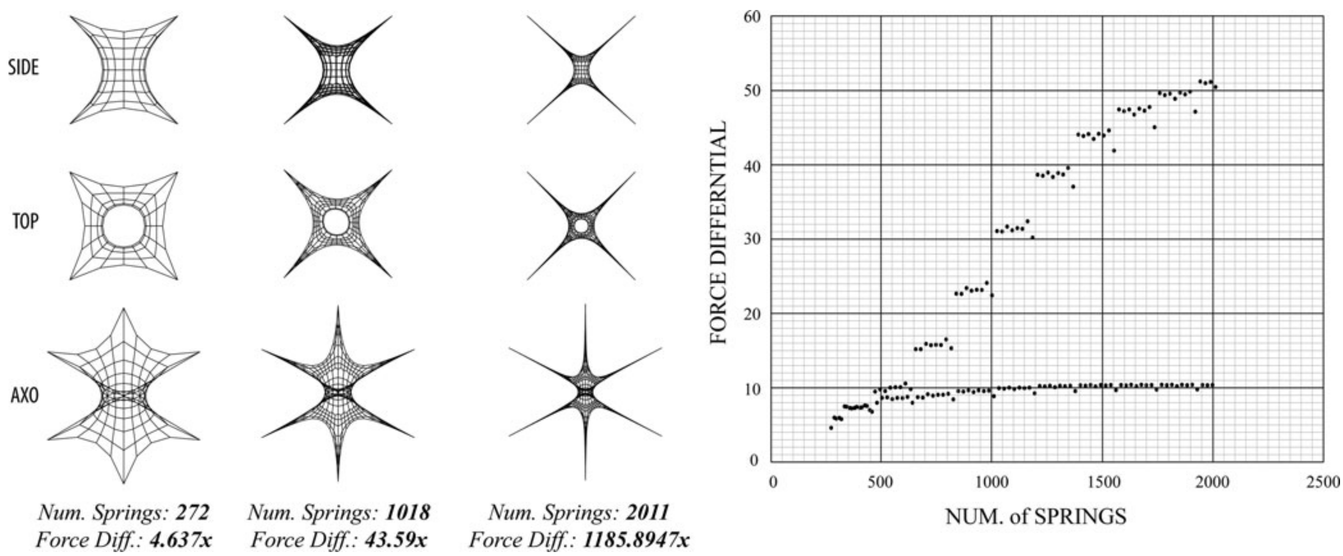

Fig. 8. (Left) Starting with a crude mesh, refinements are introduced step by step at each anchor point. (Right) The graph shows the constant fluctuation in performance until all boundaries are resolved.

the research focuses on the necessity to introduce methods of evolutionary design to navigate such an expansive design space in a comprehensive and informed manner.

The springFORM software program is tailored to allow manual manipulation of topology during the form-finding process. Yet, individual manipulations significantly alter the performance and form across an entire interconnected mesh, as previously described. At a certain level of topological complexity, visual or even numerical examination of iterative changes may not clearly present discernable rules that underlie the relationship of a particular topological change to a certain resultant form and structural performance. Such is the challenge for engaging emergent conditions within a design context. Emergence is the process at which patterns, of a structural and/or geometric nature, arise without a discernable understanding of the constraints and forces that define such features of the system (Crutchfield, 1994). In a design context, it is the *instrumentalization* of emergent features that is of vital importance. Therefore, it is critical to enable observation simultaneously with encapsulation and exploration of the unique potentials housed in emergent features. This poses the primary impetus for developing a heuristic method for topologically complex tensile meshes in addition to the manual and deterministic means of topology manipulation in the springFORM environment.

3. METHODS

This research builds upon the basic approach of genetic algorithms (GAs) following processes of natural selection and evolution to generate systems of specific fitness within a given environment (Holland, 1992). Evolutionary strategies (ES), developed by Ingo Rechenberg along with but independently of Holland's GA, utilizes an explicit nonbinary description of the genetic code and embeds mutation operators as a part of each individual's genotype. For this research, in developing an organizational structure for mesh topologies, ES offers an important tool where the genotype contains both *strategy parameters* and *object variables* (Jones, 2002). Mutation is the primary operator, in ES, for generating new and differentiated individuals by altering the strategy parameters. The mutated strategy parameters then produce the explicit values that eventually generate the unique phenotype.

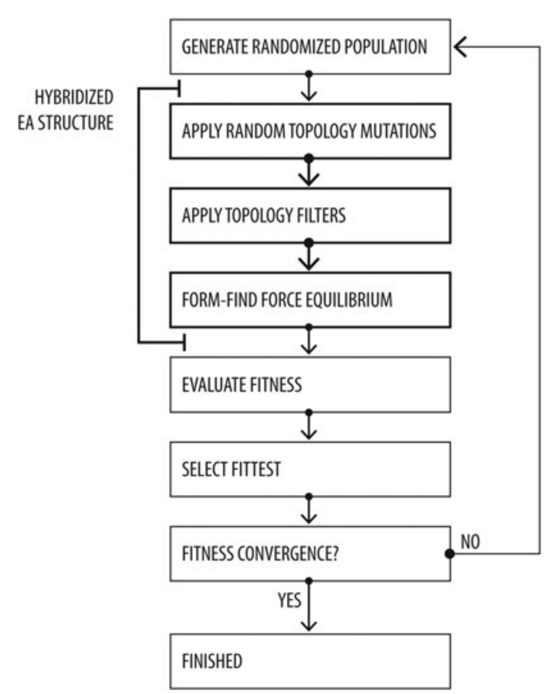

Fig. 9. Structure of the hybrid evolutionary algorithm, reliant upon the combination of mutation and deterministic methods for *filtering* topology and calculating structure through spring-based form-finding.

https://doi.org/10.1017/S0890060415000402 Published online by Cambridge University Press

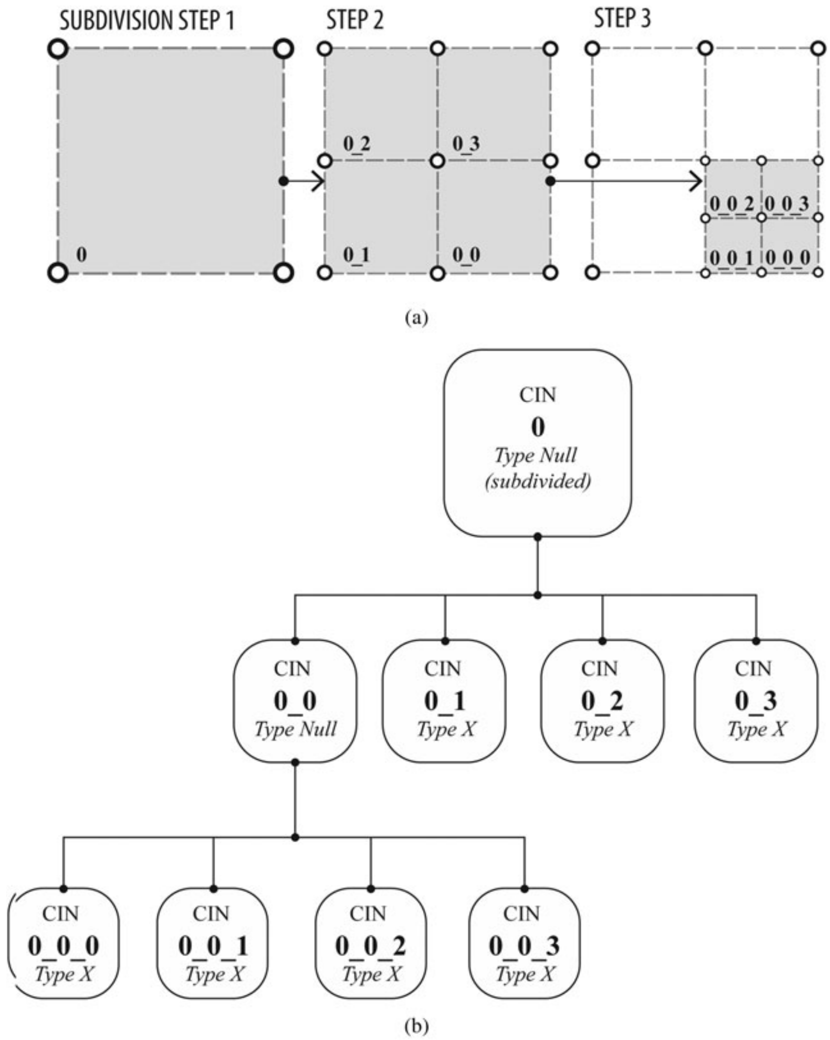

Fig. 10. Cell identifier notation (CIN) method based on the primal quadrilateral quadrisection that organizes nodal groups for insertion of a variable logic of topological (spring) elements. Three levels of nodal subdivision, shown in (a) are stored in the CIN tree-structure (b) where unexpressed levels of parent subdivision have cell type "null," meaning they contain no springs.

ES offers the key possibility for varying the actual methods for construction of an individual not just the variables inserted into a single constructor method. In addition, the logic of genetic programming influences the methods of encoding and generating complex topologies in this research, focusing on the evolution of interacting operators as mesh constructors. In genetic programming, operators are organized into a syntax tree as opposed to the typical linear list of chromosomes defining the genotype (Poli et al., 2008). This aspect plays into the organization and interactions between multiple levels of subdivision within a given mesh.

A key facet in this research is the construction of a robust mesh generator that can produce great degrees of topological variation within a single mesh structure, accounting for both variations in number of elements and the manners in which the elements are interconnected. Peter Bentley introduces the concept of *embryogeny* to define the transformation from genotype to phenotype as a process of *growing* (Bentley & Corne, 2002). The method of generation can either work as a fixed engine, an *external* embryogeny, which passes on the recombined and mutated genotype as a simple set of binaries or variables, or the method can be housed *within* the genotype and open for mutation, defined as either *explicit* or *implicit* embryongenies. The former is most commonly used in design contexts implementing mathematical means for geometry generation such as with Voronoi/Delaunay tessellation (Oliyan & von Buelow, 2014). This poses a key problem addressed in this research: when trying to simultaneously explore potentials and constraints related to materiality, such an approach assumes that an effective form can be produced with a single geometric schema, which is typically not evident at the outset (Gerber, 2012). In response, this research focuses on implicit embryongenies because they are classified as evolvable methods for growing individuals. In comparing explicit and implicit methods, explicit embryogenies use a governing set of instructions that dictate each step of

https://doi.org/10.1017/S0890060415000402 Published online by Cambridge University Press

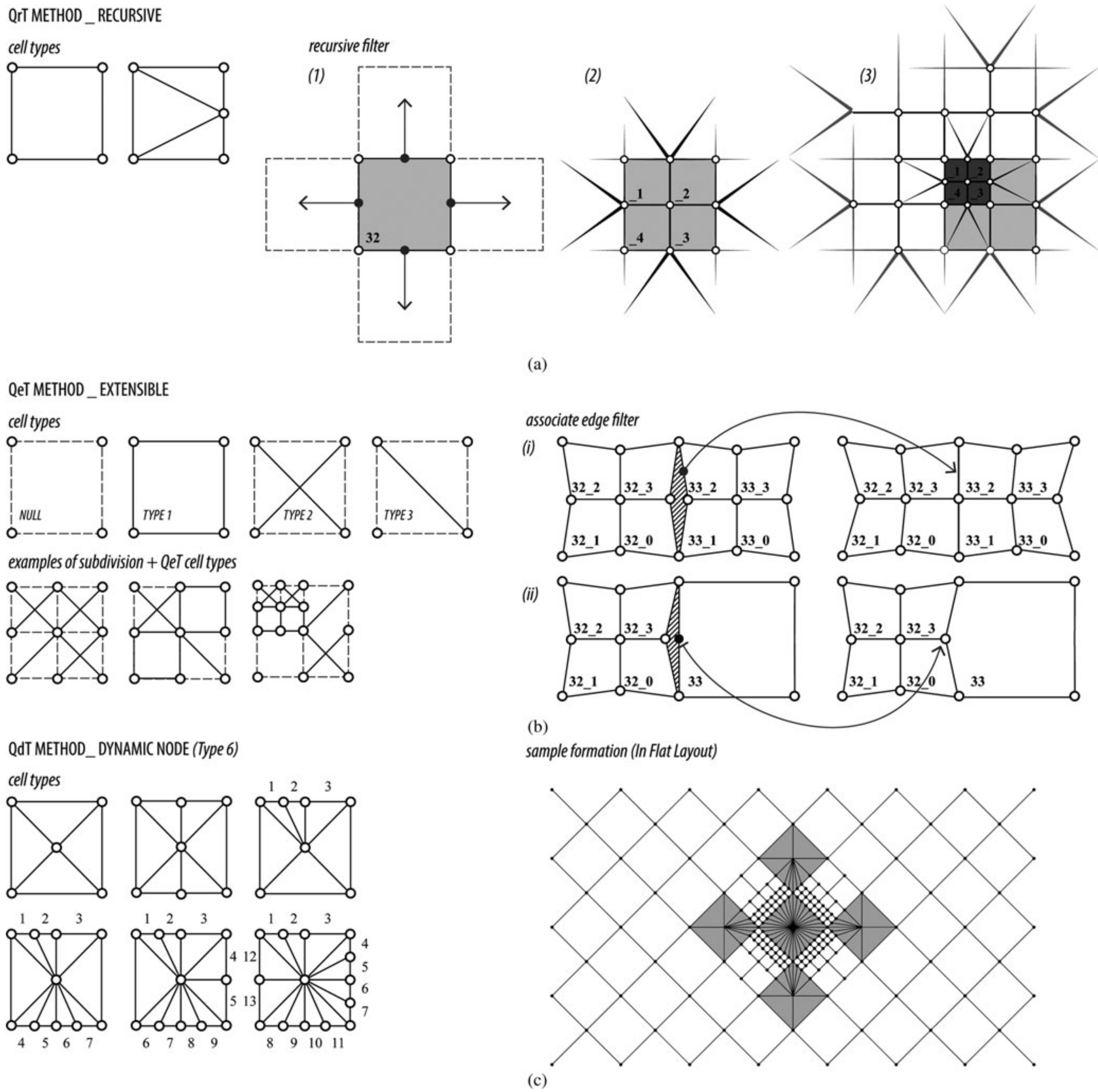

Fig. 11. (a) Recursive, (b) extensible, and (c) dynamic node quadrilateral quadrisection topological transformation methods and related filters for addressing neighboring conditions.

growth, seen in GAs with methods such as shape grammar or Lindenmayer systems (O'Reilly & Hemberg, 2007). Such methods can be problematic as the set of instructions for growth expand in length and complexity as the GA advances. Alternatively, implicit embryogenies are rule based and operate iteratively within the generation of each individual. They follow a more authentic implementation of natural evolution where the method of development "is not completely predetermined and preprogrammed, it is dynamic, parallel and adaptive" (Bentley & Kumar, 1999). Utilized with this research, an implicit embryogeny allows for great scalability and, of more importance, the ability to produce a wide variety of topological associations and geometric schema, allow the set of instructions to evolve, and the most appropriate nonperiodic solution to emerge.

3.1. Hybrid EA structure

The term hybrid EA has been applied to the common practice of intermixing aspects of evolutionary methods, shown in

https://doi.org/10.1017/S0890060415000402 Published online by Cambridge University Press

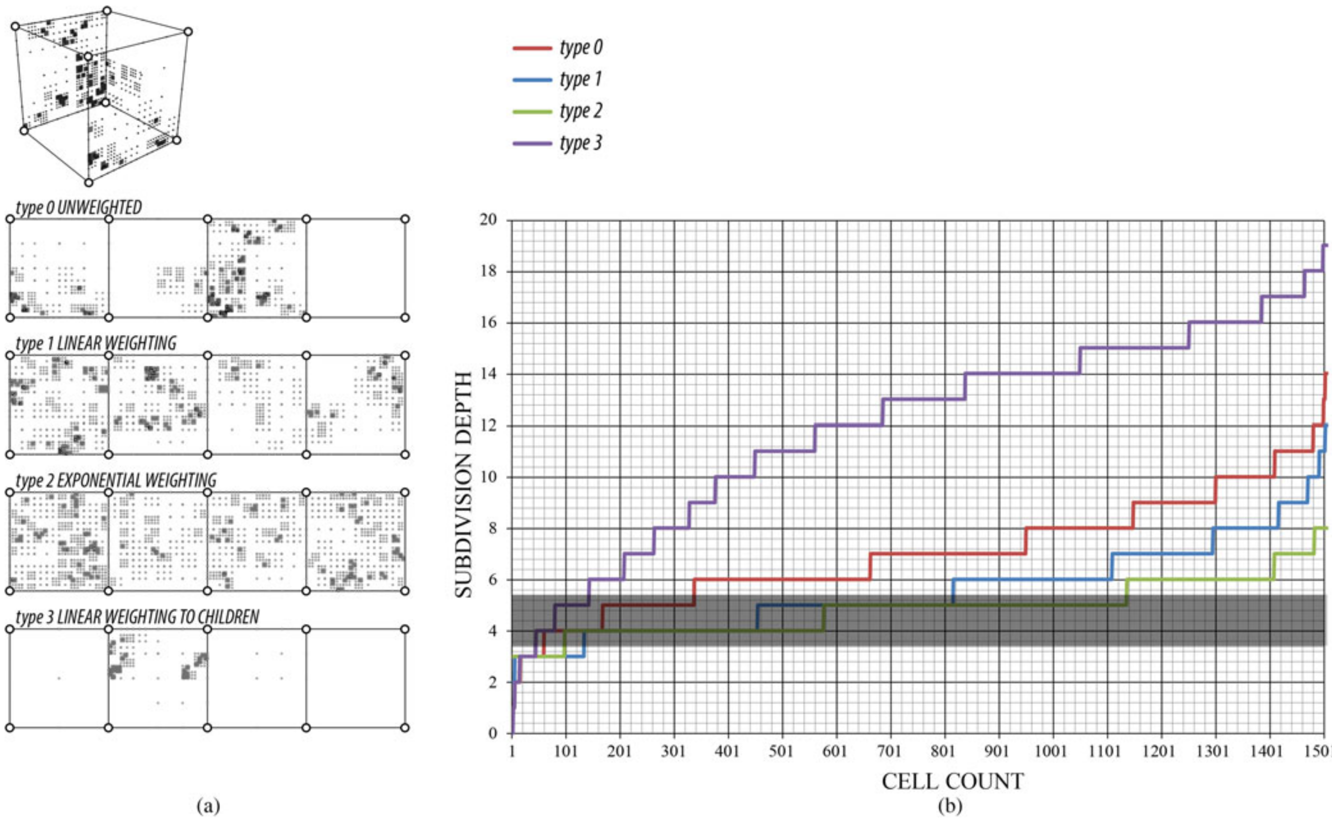

Fig. 12. (a) Schemes for cell selection: 0, unweighted; 1, linear weighting toward lesser subdivided cells; 2, exponential weighting toward lesser subdivided cells; and 3, linear weighting toward more subdivided cells. (b) The graph indicates the variation in subdivision concentration and distribution among the selection schemes.

particular with research for implementing mutation rate as a part of the genotype (Dianati et al., 2002). The use of *variable chromosome lengths* provides insight for this research in the need to grossly increase, vary, and manipulate topology through the course of evolutionary development. One approach for variable chromosome lengths uses progressive refinement to extend the length of the chromosome with each new generation, introducing new possibilities for phenotypic features (Kim & de Weck, 2005). By maintaining constant chromosome lengths within a generation, recombination and crossover can proceed in a logical manner. Where chromosome lengths are different within a population, the crossover method has to be refined, such as in research where only intraspecies breeding could occur to avoid mixing of inappropriate genes, especially in cases of using real values instead of binaries (Ryoo & Hajela, 2004). In our research, individuals of varying topological structures (as number of subdivisions) are addressed within a single population, where topology, mesh density, and location of particular topological features have to be evolved simultaneously. This research focuses on the use of mutation as the primary engine for phenotypic variation, eschewing the use of a crossover method among populations where chromosomes of potentially greatly differing subtrees exist.

With the heavy reliance upon mutation as an evolutionary engine, there is a necessity to consider adaptation as a means to limit degrees of variation and moments of severe misfits (Russell, 1992). In embryological development, adaptation can occur through means of homeostasis, where physiology drives formation, and continual readjusting to environmental influence (Turner, 2007). In the case of examining structural performance, methods for *repairing* topology can allow previously defective individuals to be structurally viable and tested for loading (von Buelow, 2007). Such methods are referred to in this research as *filters*, acting primarily to ensure structural continuity in meshes that have differential levels of subdivision. The combination of reliance upon mutation for subdivision and topology manipulations, along with filters for preventing ill-formed meshes, defines the hybrid EA structure in this research (Fig. 9).

3.2. Cell identifier notation, subdivision, and filter methods

Where mesh networks are difficult to operate on when considered in their entirety, means are necessary to localize topological operations for networks of arbitrary complexity. A cell identifier notation (CIN) method has been introduced in this research as a discretization step for topologies of any level of complexity in density, differentiation, and irregularity. By organizing nodes into operable groups, defined as *cells*, subdivision or other operations can be stored using a

https://doi.org/10.1017/S0890060415000402 Published online by Cambridge University Press

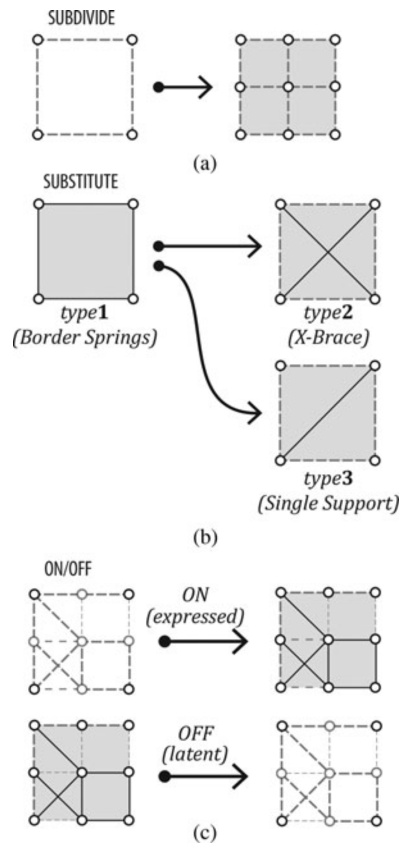

Fig. 13. Mutation steps: (a) random insertion of new subdivision, (b) random substitution between cell types (spring topologies), and (c) random switching of existing subdivisions on or off.

relatively simple tree-branch syntax (Fig. 10). Conceptually, the CIN is a tree-branch structure designed to allow for efficient querying of local cell information. It is implemented as a *linked-list* data structure where parent cells maintain references to their child cells (Sedgewick & Wayne, 2011). A subdivision operation transforms a "parent" cell to a series of "child" cells using the logic of the primal quadrilateral quadrisection (PQQ) method. PQQ is an edge-based mesh refinement method used in computer graphics for subdivision algorithms and multiresolution modeling (Shiue & Peters, 2005). Implemented within the CIN method, subdivision is considered as a step completely separate from explicit topology. Therefore, the PQQ-based CIN method is purely organizational and does not contain any geometry or considerations of geometric relationships. Data for topology is inserted as a *type* definition within each cell, allowing for great ease of switching topologies without altering the CIN structure.

The CIN is agnostic to cell *type*, which is the indicator of the topology of springs associated with the nodes of a particular cell. Within this system, a set of nodes can be connected in any fashion, which is how the CIN differs from the traditional concept of a subdivision scheme. At its most basic level, the CIN acts as a locator of a set of nodes that are cognizant of neighboring conditions. By separating topology from cell identification, when a cell is subdivided, its spring topology is deferred to the children cells, but its location still exists within the CIN method, thus allowing for expedient search through the mesh network. The implementation of the CIN also embeds neighbor relationships at the node (particle) level. These neighbor relationships make it easy to traverse across the cell graph once a nodal location has been established.

While the CIN method is open to any spring topology at the level of each cell, three primary cases for topological transformation within the quadrilateral quadrisection method have been developed: recursive (QrT), extensible (QeT), and dynamic (QdT; Fig. 11). Because it has been shown that the CIN method is able to embed and traverse neighbor conditions, each of these methods also involves *filters* that manage such neighboring relationships after spring topology has been inserted with the cell. The filters are deterministic engines that resolve neighboring conditions with disjunctive levels of nodal subdivision. The QrT method inserts *support springs* that resolve neighboring cells where there is a difference of one level of subdivision. The recursive method will introduce new subdivisions so that the rule of neighbors having only one level of subdivision difference will always be met. The QeT method introduces a wider array of cell topologies, examining cell edges and stitching them, where levels of subdivision differ. This method accommodates neighboring cells with any number of differences in the levels of subdivision. The QdT method allows any topological description to occur within the cell to provide complete interconnectivity with the subdivisions of neighboring cells. Therefore, there is only one explicit cell type for the QdT method, although its topological description can greatly vary.

3.3. Weighted selection and topology mutation methods

The method of cell selection within the tree-based CIN has a number of inherent probabilistic biases that can influence the efficacy of the EA. With the reliance of evolutionary methods on stochastic sampling, it is necessary to address how random sampling might influence the exploration process. Given that any PQQ-based subdivision operation has the effect of introducing new cells into the system, registered as an increase in *depth* in the CIN tree, a direct relationship is noted between the search algorithm for selection and the process of topology modification. Specifically, given a quadrilateral subdivision strategy, a single split operation increases the number of cells in the system by three; the parent is replaced by four new cells.

$$\text{Cells}_{T+1} = \text{Cells}_T + \left(\sum_{i=0} \text{Cells}_{\text{Introduced in subdivision}} - 1 \right).$$

As a by-product of the tree-branch structure of the CIN, a simple unweighted selection scheme will favor a more rapid localization into already subdivided areas with a bias directly relational to the number of new cells created by a single subdivision operation. To make these biases operable by the al-

https://doi.org/10.1017/S0890060415000402 Published online by Cambridge University Press

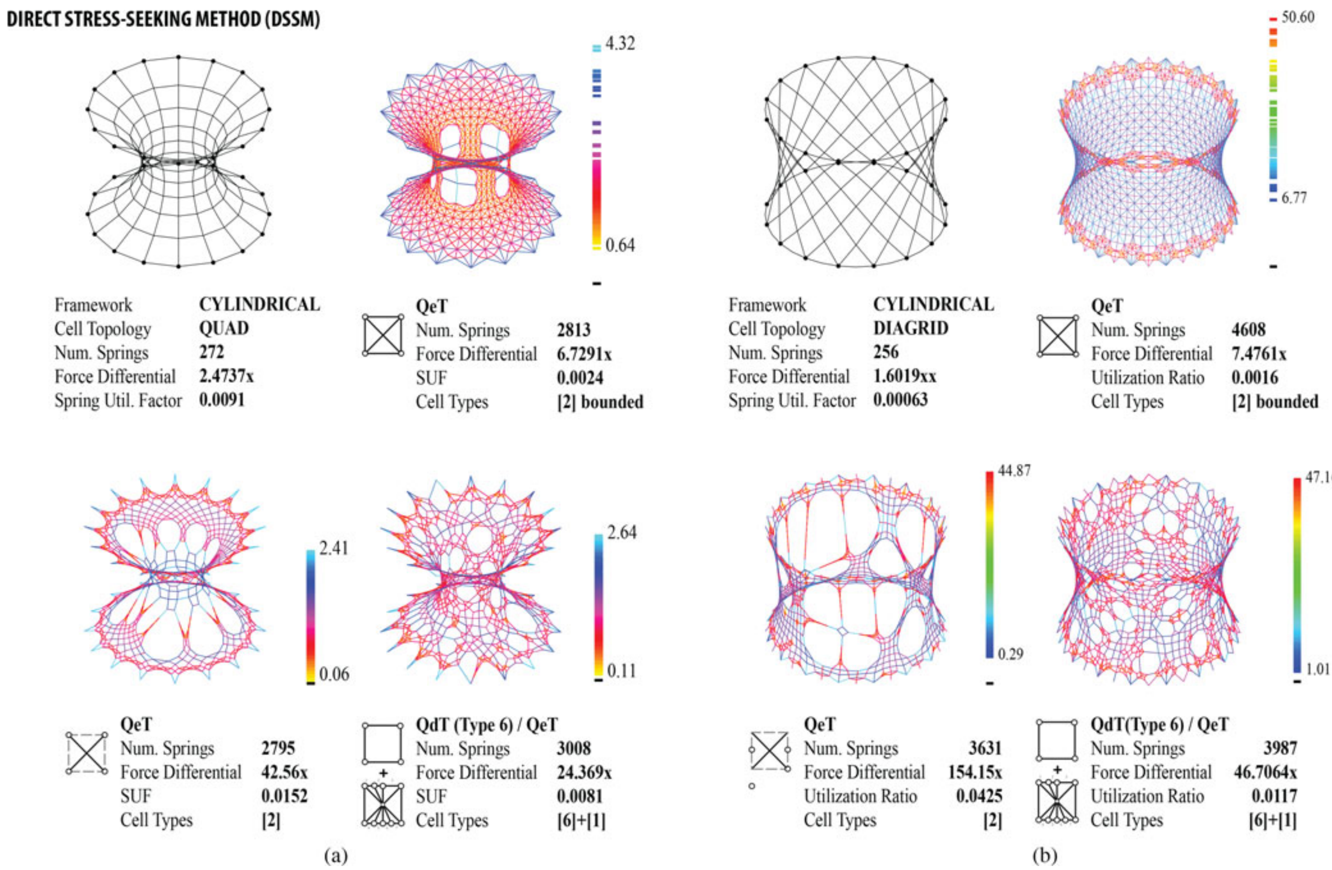

Fig. 14. The direct stress-seeking method applied to a simple (a) *quad-grid* and (b) *dia-grid* cylindrical mesh, utilizing extensible and dynamic quadrilateral quadrisection subdivision schemes with varying cell types.

gorithm itself, a set of cell selection methods are introduced, allowing for the precise tuning of sampling that balances breadth and depth for searching through the tree-branch data structures (Fig. 12). The unweighted scheme, *type0*, shows how a top-level cell can remain completely unsubdivided from its initial state, after a run of 500 subdivisions have been randomly applied. In applying weighted selection methods, the linear, *type1*, and exponential, *type2*, show a concentration of subdivision depth in the graph, a preferable result where distribution of subdivision is relatively even and the range in levels of subdivision is minimal. On the other extreme, with a weighted selection toward subdivided cells, *type3*, shows intense localized subdivisions, where the range of subdivisions levels is extremely high and deep.

3.4. Mutating cell subdivisions

The EA in this research relies upon three separate mutation steps to control topological variation of springs within any given population, where mutations are located based upon the selection methods defined previously. The genotype of any individual is defined by the CIN tree and a topological cell type applied to each cell within the CIN tree. Mutation is the application of three randomized steps to selected cells: introducing new subdivisions and assigning cell types, switching cell types among existing cells, and turning existing cells on or off (Fig. 13).

The first mutation step is the insertion of new subdivisions and the selection of cell type for these newly generated cells. These cells are immune to the other mutation functions in the generation where they are created, so that they are allowed to express themselves in the phenotype. The second mutation step is the ability for any cell to switch between available cell types. When the algorithm locates an area within the mesh that should receive an operation, the type switch allows for the algorithm to try a variety of possible force direction and mesh resolution options at that location. This gives the algorithm a much finer-grained control over the difference between location and type of topological operation. The third and final step allows any given subdivision operation to be turned off or on at a specific location. In this way, certain features can either be expressed or nullified in the resulting force network. This provides means for both generative and degenerative growth, where subdivision or mesh density can be increased or decreased in specific areas. The turning off of a cell is stored in the genotype, allowing it to be reinstituted and expressed again later in the evolutionary process.

https://doi.org/10.1017/S0890060415000402 Published online by Cambridge University Press

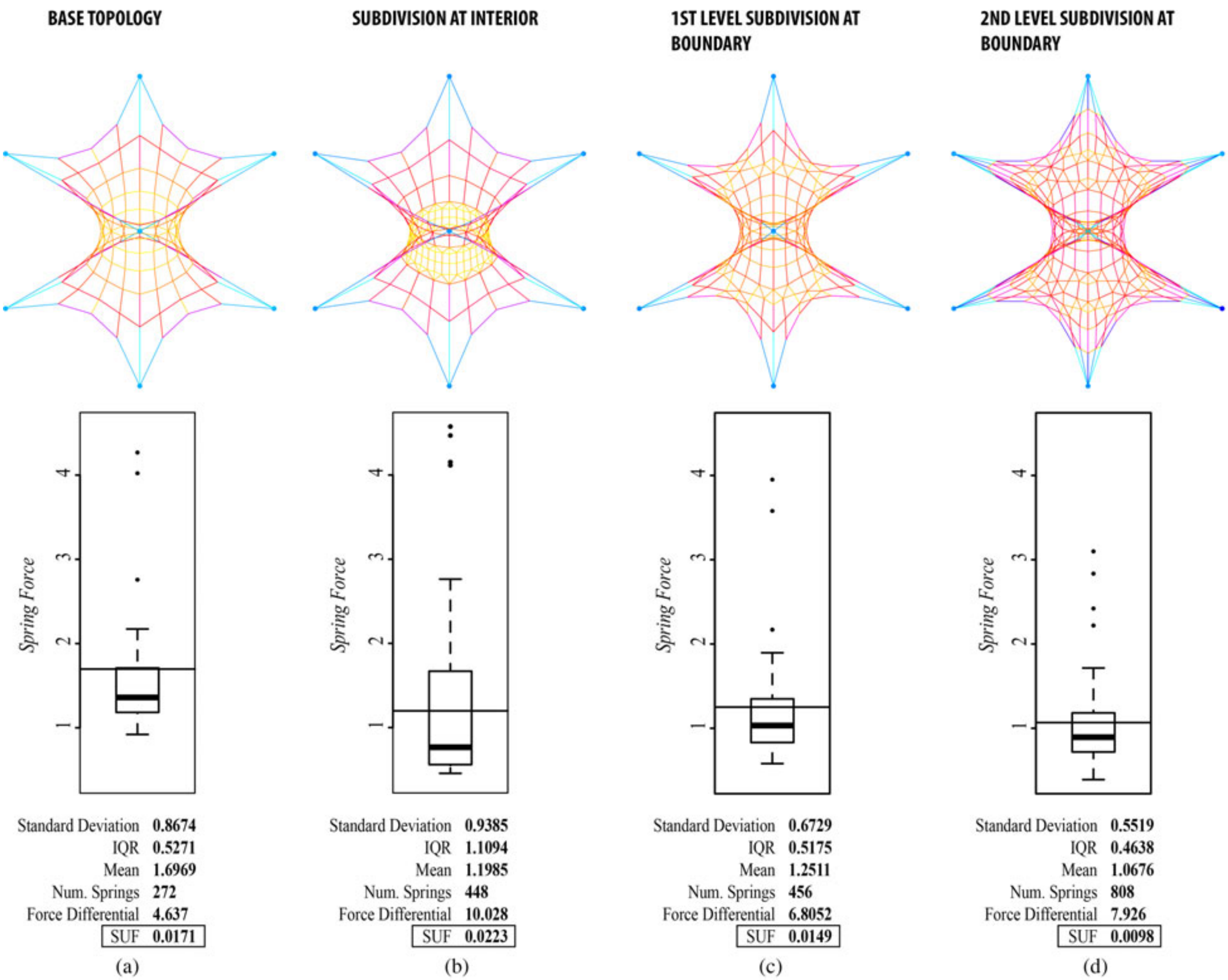

Fig. 15. Comparison of various statistical measures for the relationship of spring count and force differential, where the spring utilization factor shows as the only method where the value continually improves (decreases) with overall density.

3.5. Direct stress-seeking method for selection weighting

The selection methods introduced above can be considered "blind" to the actual phenotypic manifestation of force networks within a form-found mesh. They are statistically oriented, where the mesh is searched holistically and without direct regard to local or specific features. To study particular emergent features related to stress distribution and concentration, another selection method is introduced that weighs cell selection based on *internal force* within the springs that it contains. This method, therefore, differs from the original selection schemes in that it is dynamic and applied *after* form-finding. It is a trade-off in that it can double the runtime of the entire algorithm (by requiring its own separate form-finding step) but can also very quickly localize topological modifications, and provide means for exploring emergent patterns and behaviors.

The direct method is a naive method of exploiting knowledge of force distribution in a form-found mesh to direct local subdivisions as a means of resolving peak stress. This provides an expedient testing ground for cell selection methods and topological cell types to indicate possibilities in emergent geometric and stress patterns. The algorithm seeks out the cell with the highest inner force by calculating the summation of all forces in the springs of each cell. The identified cell is subdivided, a subsequent form-finding step is processed, and the routine is repeated. Asymmetries occur because the insertion of a new set of springs within one cell redistributes forces and thus produces a different set of values for the inner force of all cells. This continues for a discrete number of steps until a termination condition is met, typically set to a desired ratio of force differentiation to number of overall springs. In comparing cell types (the type of subdivision utilized) and the underlying initial grid, whether quad-grid or dia-grid, features of mesh organization, density, pattern, and level of success in minimizing the force differential are identified (Figs. 14 and 15).

3.6. Spring utilization factor

Looking at standard measures of mesh articulation and structural behavior, there are two primary categories: global or statistical measure, and local or feature-based measure.

https://doi.org/10.1017/S0890060415000402 Published online by Cambridge University Press

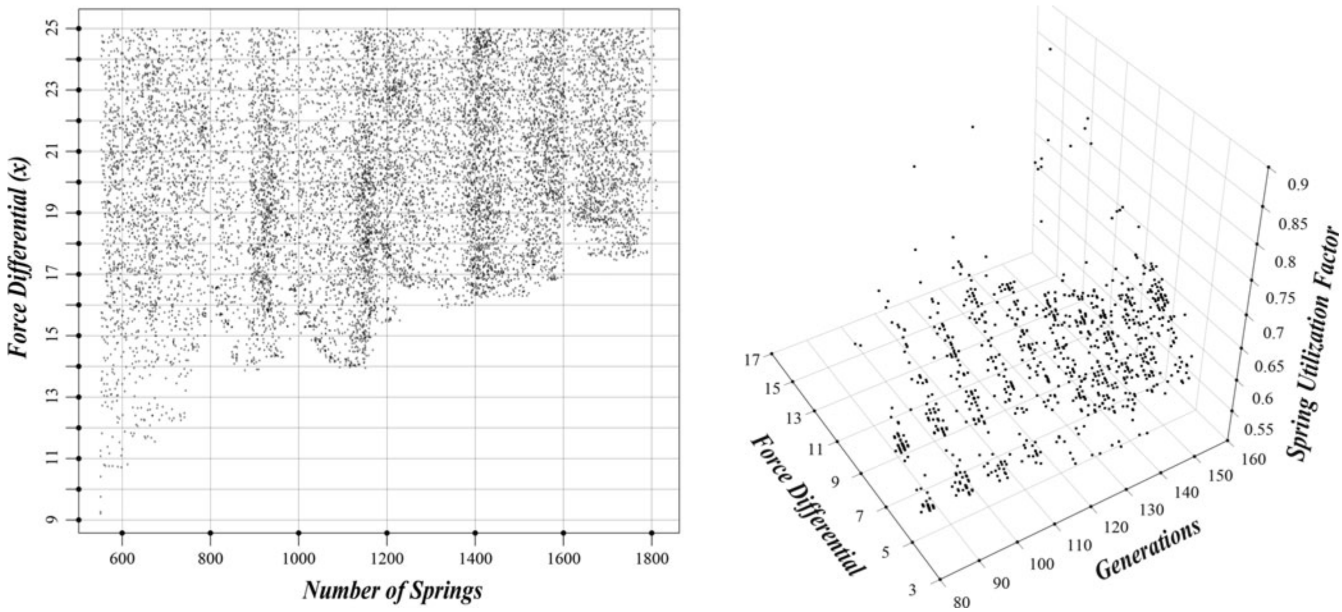

Fig. 16. Pareto front for the relationship of force differential (subset with values <25) and number of springs, showing the inevitability of increasing force differential as more spring elements are introduced into the system.

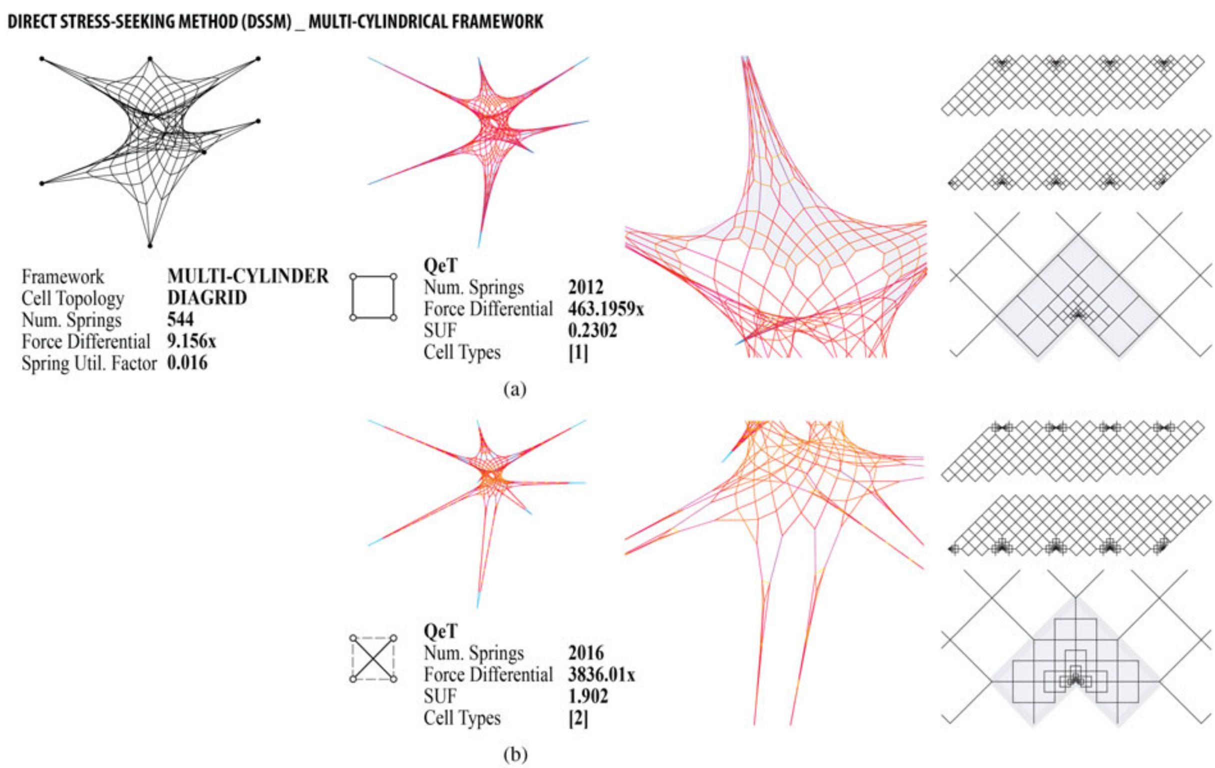

Fig. 17. Samples using direct stress-seeking method with multicylinder topology.

Statistical measures of the structural members, or the aggregated properties of the springs themselves, are descriptive but fail to capture the variation of local behaviors and properties. As one example, when assessing the standard deviation of spring forces, the algorithm for subdivision can "cheat" by adding more and more springs in the neutral areas of the mesh. By adding only minimally stressed springs, the density is increased and the standard deviation is reduced, but the poles are not addressed in terms of the maximally and minimally stressed springs. Methods of averaging that assess at the global level of the system produce similar results in not being able to address and minimize force differential (Fig. 15).

https://doi.org/10.1017/S0890060415000402 Published online by Cambridge University Press

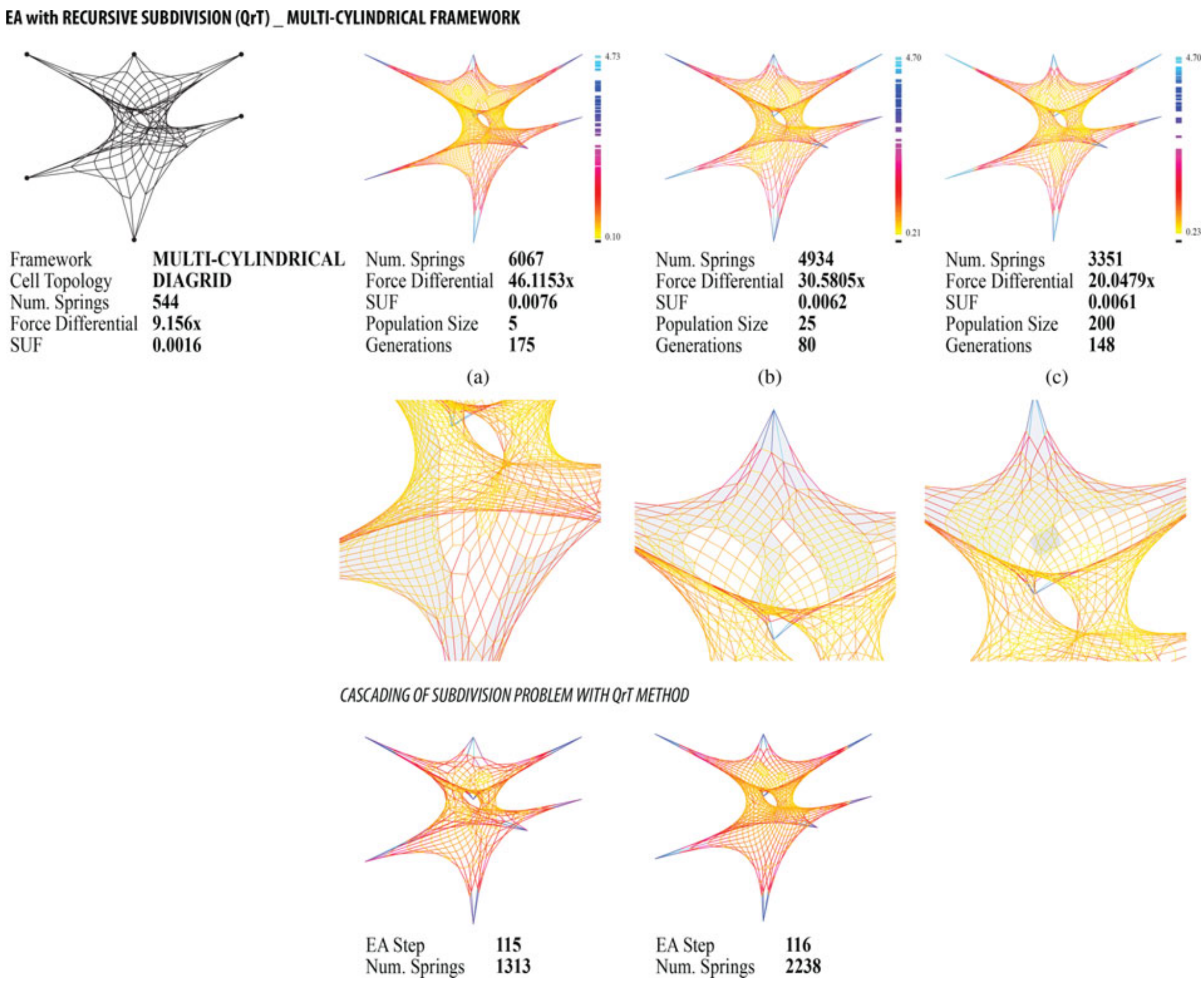

Fig. 18. Samples from multiple evolutionary algorithms (EAs) runs using the recusive quadrilateral quadrisection method while varying population size and number of generations, showing (bottom) an example of the cascading of subdivision occurring across only one EA generation.

Adding density to a mesh comes at a price, defined structurally as an increase in relative force differential between the least and most stressed members. With the addition of springs into the system, the force differential inevitably increases as there are more elements exerting force. The desired outcome is a uniform force distribution, or a minimization of force differential, while understanding that, at the same time, additional members precipitate disequilibria. In simple terms, the Pareto front for desired mesh resolution and topological differentiation is defined between two dimensions: the number of springs in the system (the "density") and the force differential within the system (the "constructability"; Fig. 16).

The rate at which these features change has been defined as the spring utilization factor (SUF):

$$\mathrm{SUF} = \frac{\Delta\,\mathrm{forceDiff}}{\Delta \mathrm{NumOfSprings}}.$$

By assessing the force differential, a "local" description is included in the fitness, looking at the specific springs that sit at either end of the scale in terms of force exertion. The SUF still functions to find the arrangements for a particular number of springs that returns the lowest factor of force differential to mesh density. In this sense, the SUF provides a measure to negotiate the desire for increased density while maintaining stresses at which a resulting mesh can be materialized by a single material type. It is important to reiterate that the interests of this research are not solely structural; the imposition of mesh density is significant. Therefore, the SUF fitness function does not punish increasing force differential; rather, it assesses a comparison of all meshes composed of a similar number of springs. Within the structure of the hybrid EA, the best performing mesh within a population is selected, even if it is a lesser performing individual than its parents (Laumanns et al., 2000). The mutation method continually introduces subdivision to achieve increasing densities, where the SUF serves as a measure for comparison and ultimately a cutoff to determine a point at which the force differential exceeds the ranges of stresses which a cable material can support.

https://doi.org/10.1017/S0890060415000402 Published online by Cambridge University Press

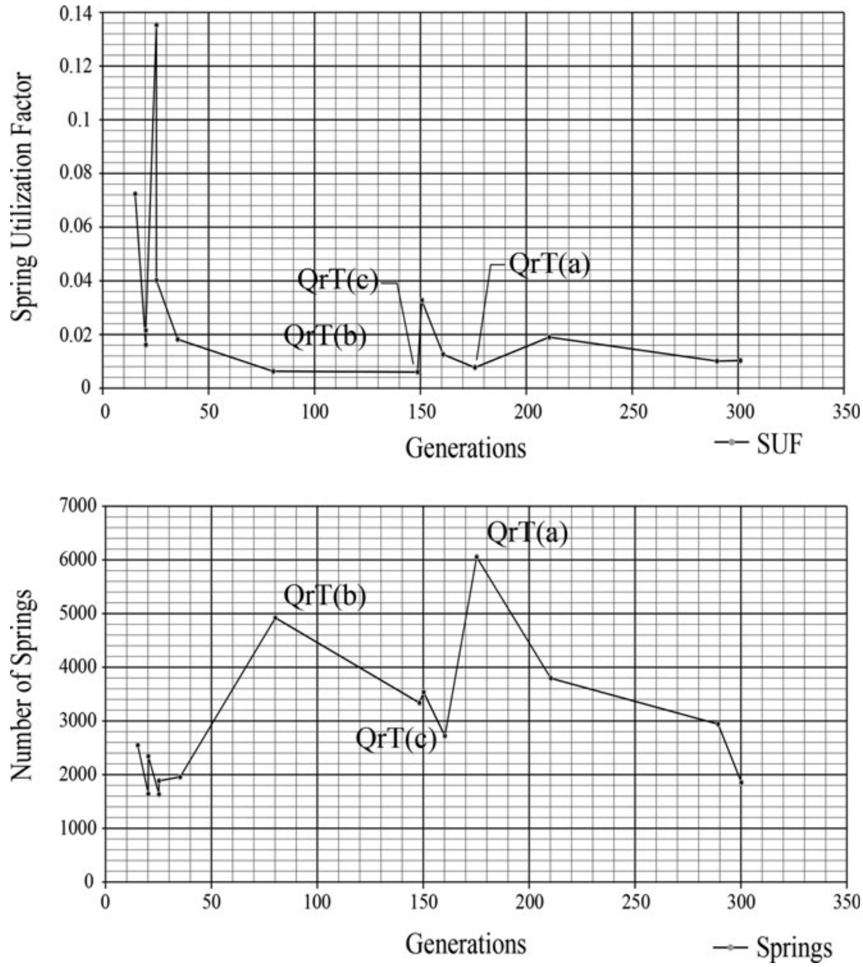

Fig. 19. Examination of results related to number of generations, showing efficiency in the recursive quadrilateral quadrisection method to produce high spring count, while the extensible quadrilateral quadrisection method greatly varies in results based upon selection scheme and cell types that are utilized.

4. RESULTS

4.1. Direct stress-seeking (DSS) method applied to multicylindrical mesh

The following tests utilize the DSS method and base *dia-grid* topology as constants, with the subdivision methods and cell types being varied (Fig. 17). As mentioned previously, the DSS method weighs cell selection to choose the ones with the highest inner force, blind of subdivision level. It therefore focuses subdivision to the anchor points, which in these exemplars are more prominent because only a minimal number of points on the boundary are fixed. The value of *force differential* greatly varies between each exemplar, where most are well beyond the threshold in which a single material could manage the vast ranges of prestress. Exemplar (b) utilizing QeT with cell type 2 returns a force differential of 27.5 through developing a bundling of springs toward the anchor points, rather than direct densification as seen in the other tests. This is a unique feature that emerges in the EA exemplars, although in this case, it produces an undesired geometric narrowing of the surface structure at the anchor points. Applying the DSS method to the multicylindrical mesh shows the limitation of a deterministic selection method, but it provides an understanding of emergent topological and geometric features for each quadrilateral quadrisection method.

4.2. Hybrid EA applied to multicylindrical mesh

Initial tests with the hybrid EA method utilize recursive subdivision (QrT), the basic quad cell type (type1) and the *exponential* selection method to aid in distribution of subdivisions throughout the mesh (Fig. 18). The primary variable in these tests is the number of individuals per population, leading to differences in the number of generations needed to accomplish convergence. In comparison to the use of the DSS method, the hybrid EA shows the ability to reach desirable force differential values, especially when given an appropriate number of individuals per population size. With 200 individuals per population, a force differential of 20 was reached in 148 generations. These examples also show the necessity for increasing

https://doi.org/10.1017/S0890060415000402 Published online by Cambridge University Press

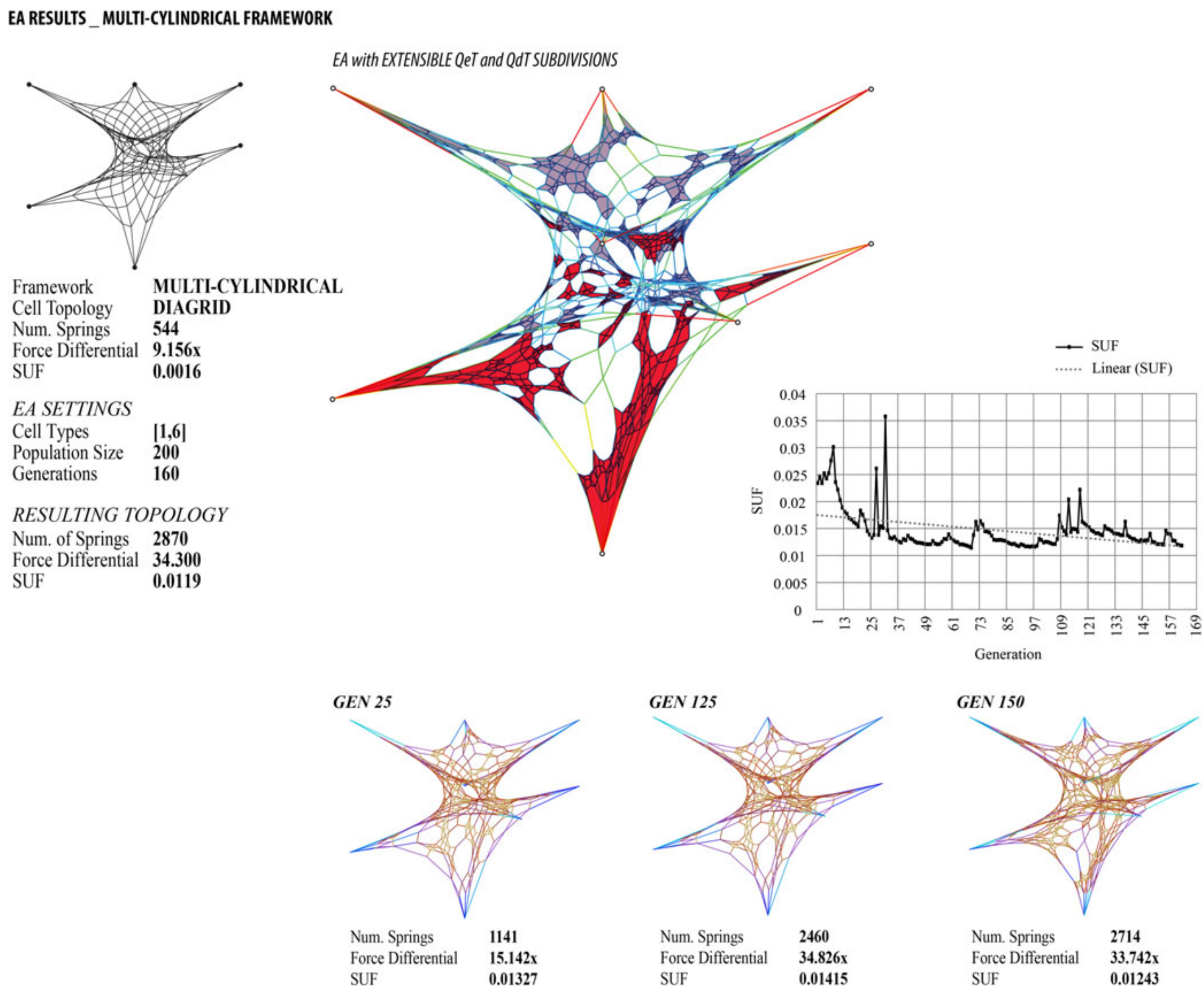

Fig. 20. Results from a single evolutionary algorithm run, showing population diversity while still maintaining particular features through the course of development, utilizing cell types 1 and 6, and population size of 200.

population size in order to explore the design space thoroughly, though this is at a cost of needing a larger number of generations. The primary difficulty with this scheme lies with the recursive action of the subdivision. As mentioned previously, the method only allows for a maximum difference of one level of subdivision between neighboring cells. The recursive filter steps through the topology after new subdivisions have been inserted to enforce this rule. What happens both throughout the generations and at the moment of convergence in the EA is that the recursive method triggers an equal level of subdivision to cascade across the entire topology. When a certain arrangement of differentiated subdivision has occurred, this cascading effect can be triggered by the subdivision of just one cell, where the knock-on effect of resolving the levels of subdivision at neighboring cells triggers all cells to subdivide to the same level.

Contrary to the QrT method, utilizing the QeT and QdT methods with no recursive filter generates a vast array of differentiated topological organizations, allowing for larger jumps in subdivision between neighboring cells. Starting with the same base topology of the tests in Figure 18, the QeT/QdT tests are run with variable cell types, population sizes, and number of generations. The hybrid EA in this instance is used as an exploratory tool where the fitness of force differential defines the fittest individual within each generation, where some tests apply a termination condition of an ideal SUF and others are allowed to continually "grow" by adding subdivisions through successive generations. The pace of growth is varied significantly across these examples, evident in the comparison of spring count to number of generations (Fig. 19). This is driven by the variation in the number of cells that could be selected for subdivision. The exam-

https://doi.org/10.1017/S0890060415000402 Published online by Cambridge University Press

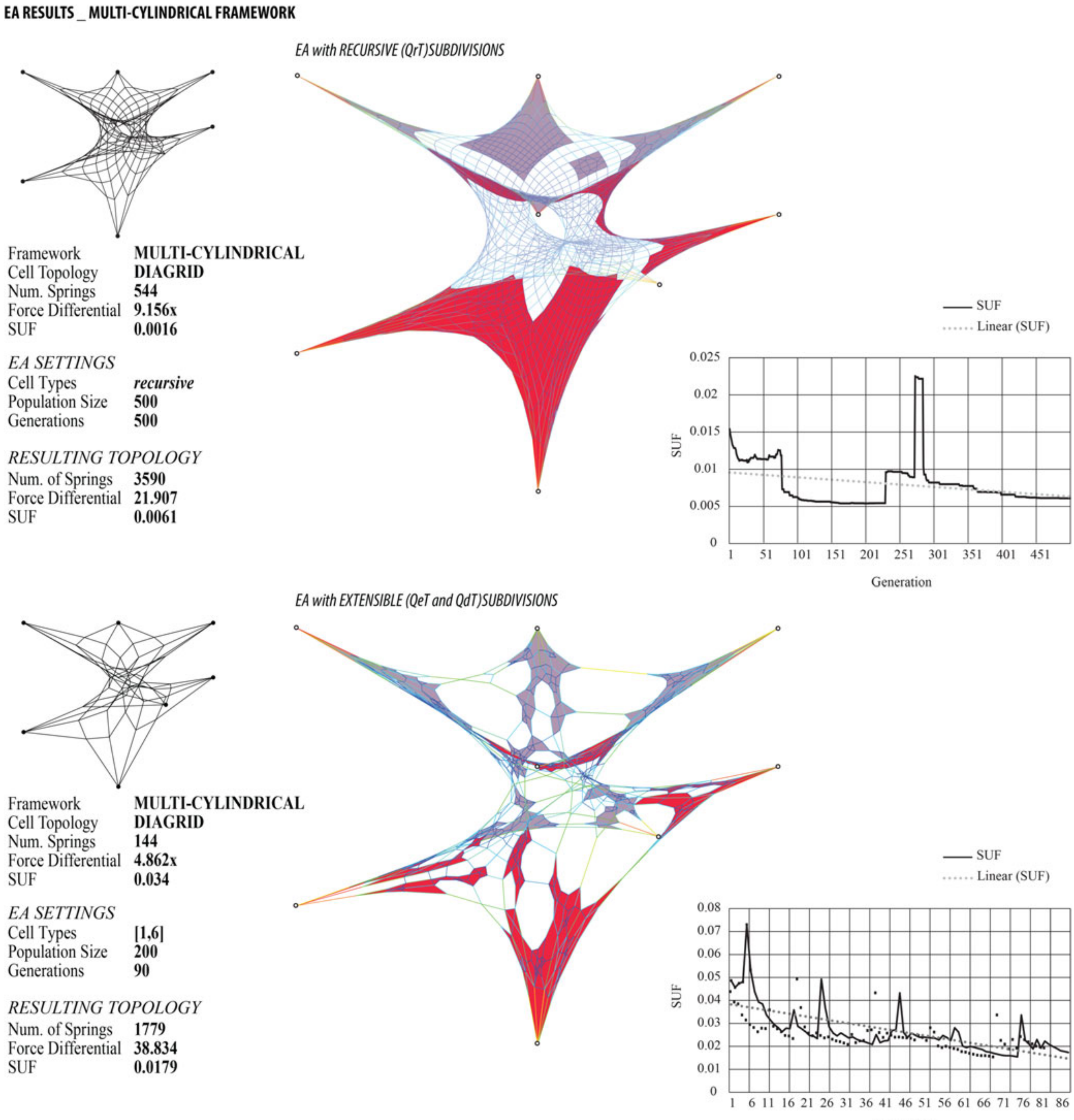

Fig. 21. Evolutionary algorithm (EA) with recursive quadrilateral quadrisection subdivision (top) showing increased population size and number of generations, where fluctuations in the fitness are due to moments of cascading subdivision. EA run using low-res, simplified base topology (bottom) with cell types 1 and 6, and population size of 200.

ples with high spring count to low generation number utilize a *percentage*-based function for determining the number of cells selected for subdivision with each individual. Thus, as growth continues, the number of cells for subdivision increases, allowing for fewer generations needed to produce a certain level of mesh density. The other examples have a discreet value for number of subdivisions allowed in each step of the EA. Ultimately, when comparing these results, no discernable trend can be identified because allowing the percentage-based selection method produces results that quickly satisfy the desire for mesh density, but the resulting fitness for spring utilization greatly varies. This can be attributed to the *expanding* design space that the percentage-based selection method produces as growth in number of cells advances.

https://doi.org/10.1017/S0890060415000402 Published online by Cambridge University Press

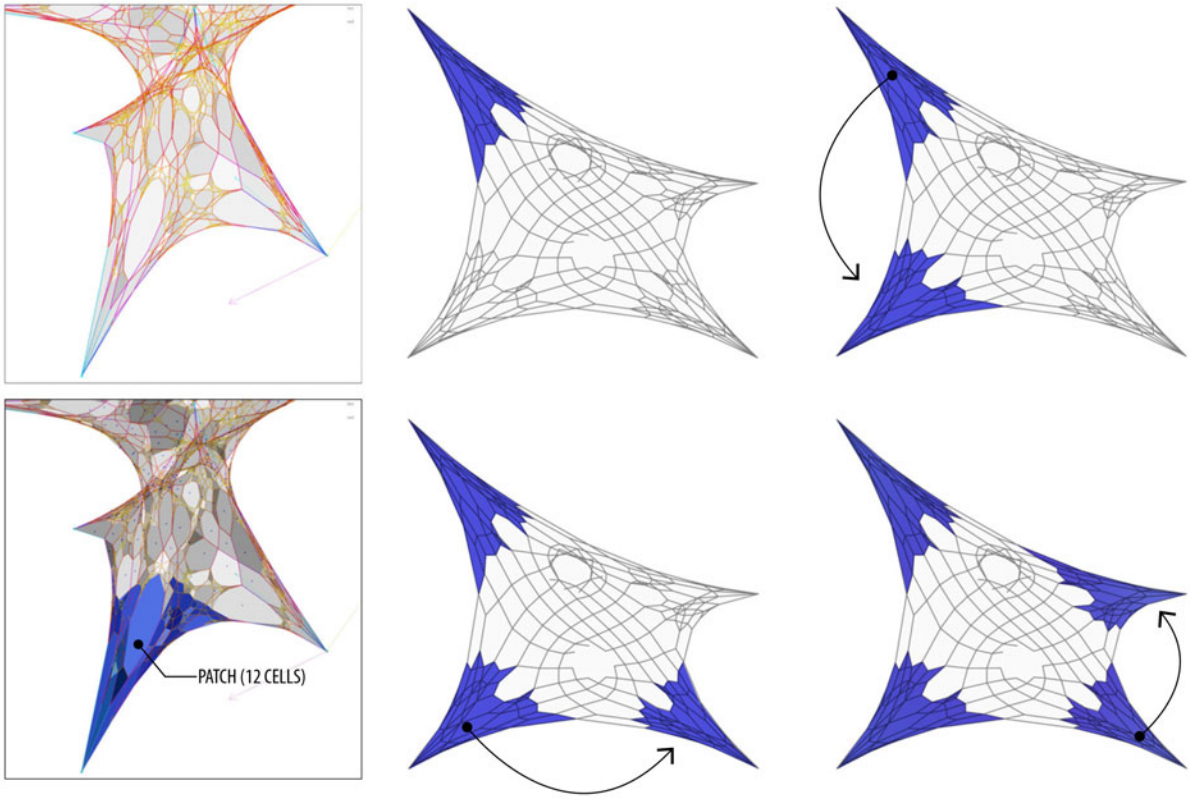

Fig. 22. Selection of underlying cells at multiple levels of subdivision to identify a patch, and subsequently grafting the patch into the other anchor point locations.

A notable distinction can be seen in comparison to the recursive QrT method, which is able to produce higher mesh resolution in a similar number of generations as the QeT method, due to the recursive action of maintaining appropriate levels of subdivision for neighboring cells.

When examining an individual run of the hybrid EA, a great degree of diversity can be seen across all the populations generated (Fig. 20). Yet it can be seen across a high number of generations that certain features, particularly at the anchor points, are maintained despite the lack of a crossover method to reinforce good characteristics. The general trend for fitness, the SUF, improves. Momentary worsening of the fitness value can be attributed to the condition described in Figure 8, an improvement in one local condition that worsens the overall performance until all local conditions have been equally resolved. Similar results are seen using the recurvise QrT method, as well as in starting with a low-resolution cell topology (Fig. 21). With the QrT method, increasing the population size and number of generations helps to produce a rigorously patterned mesh, where subdivision is concentrated at the nodes. However, the fitness graph shows how moments of cascading subdivision, as defined in Figure 18, quickly diminish the overall performance. The topology of the input mesh, defining the number of based *cells* for subdivision, shows a significant influence in the development of topological and geometric features. With a more low-resolution topology, greater diversity in mesh concentration can be achieved, where removing subdivision from a top-level cell expresses a significant void in the connectivity. The hybrid EA tests that utilize the QdT method, combining a simple quad cell with the dynamic variable topology cell, produce the most diverse meshes in the ability to generate *n*-sided polygons. Such geometries are not results of the topology of a single cell, but rather the results of a series of interconnected neighboring cells that when stressed, unfold into multisided *metacells*.

4.3. Grafting method for operating on EA results

Although the generative DSS and EA methods both operate with mass-spring networks at the level of the *cell* through the use of the CIN method, the cell itself has very limited value as a unit within the design process. Rather, the emergent features are clearly manifestations of forces applied through spring aggregations developed by collections of particular cell types and levels of subdivision. This represents an important feature of the research in the ability to freely produce great varieties of spring topologies, yet maintain a logical and accessible ordering and data structure. What is identified in the EA runs with the QeT and QdT methods are great varieties of topological structure *within* each individual. Most readily apparent are the myriad ways in which forces are resolved at the anchor points, where peak stress ac-

https://doi.org/10.1017/S0890060415000402 Published online by Cambridge University Press

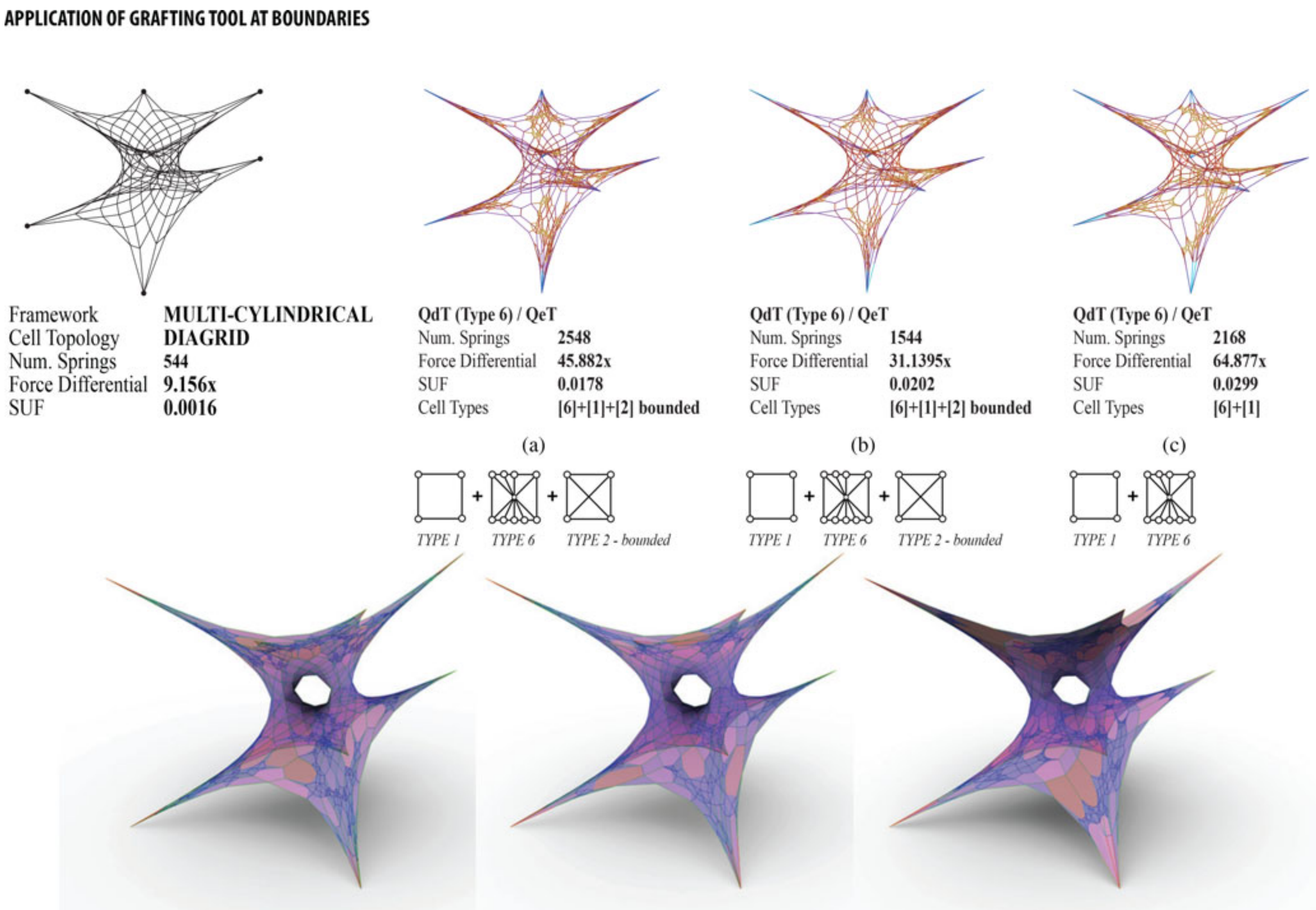

Fig. 23. Grafting patches from several evolved spring meshes into a new base topology and arraying to all anchor point conditions within the mesh.

cumulates. Inherent in the undirected nature of the EA, a single evolved individual would resolve these anchor point conditions with a range of different methods for bundling stresses. While this produces ideal densities and efficient resolution of stresses, the significantly differentiated nature of the emergent patterns is not desirable.

Positioning the EA method as an exploratory tool, a method is introduced to be able to work at a metalevel of cells, referred to in this research as *grafting*. This method looks to progress from operating at the level of data structure, which is what the EA does, to working at the level of the phenotype of the whole system. This provides the necessary access to evolved emergent features without the additional overhead of further manipulating the EA's complex mechanisms (Welch & Witkin, 1994; Schmidt, 2012). The grafting tool works in springFORM to initially define collections of cells and identify their associated subdivision operations, in the levels of subdivisions and cell types as stated by the CIN. The identifier for the top-level cells is swapped with just a directional logic for neighboring, in order to make the operations agnostic to location. The method then uses the higher order *instructions* in order to graft the set of cells onto a new location within the same mesh structure (Figs. 22 and 23). The grafting operation ultimately encapsulates both topological data and emergent geometric features into a single set of instructions. The use and operation of the grafting tool is itself an essential part of the hybrid EA method, as a productive part of the design ideation and iteration process. Because springFORM is a tool for quickly articulating functional, tensile structures, grafting is likewise designed for rapid application of multiple different functional design alternatives *into* the model, at the scale of regions of meshes.

5. CONCLUSION

The hybrid EA developed in this research has been shown to exhibit great exploratory possibilities in combination with a constrained search for highly differentiated tensile meshes. The method expands the potential for a computational process to overcome the challenges of preplanning for topologically complex models. SpringFORM, as a tool for the generation and manual manipulation of prestressed mass-spring networks, allows for intuitive and tacit knowledge to be explored in an instantaneous manner with minimal to no overhead for preplanning of specific geometric and material properties. Expanding springFORM and embedding the hybrid

https://doi.org/10.1017/S0890060415000402 Published online by Cambridge University Press

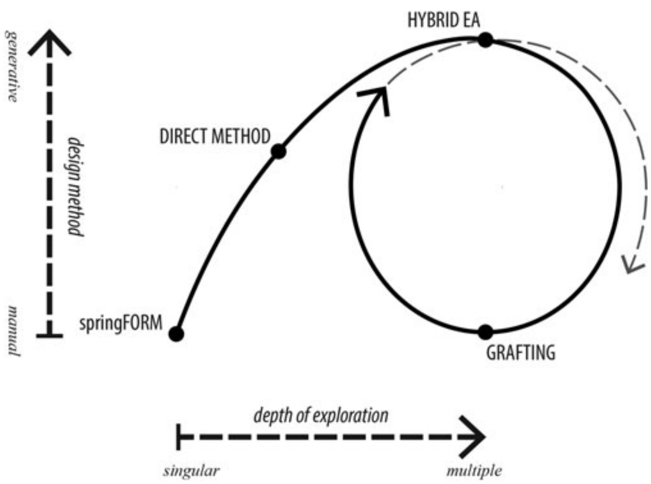

Fig. 24. Diagram of design framework able to advance the depth of exploration while maintaining ability for "manual" intervention.

EA stretches this mode of design to explore greater depths of topological differentiation and assess the possibilities of integrated spatial form and structural performance. This follows an established trajectory for managing design of increasing complexity through deploying increasingly integrated, associative, and algorithmic processes (Aish, 2013). The research presented here introduces an innovative step of returning these heavily automated methods into simple manual controls, through the use of the grafting method (Fig. 24). Reiterating the conceptual approach of springFORM, manual manipulation is accessible at advanced levels of design complexity to foster continued development of tacit knowledge. Grafting allows for topological manipulation to function as a logical means for design exploration because it operates with the embedded knowledge of the hybrid EA.

Continued development focused on this concept of overlaying means of manual design engagement with methods of advancing computational complexity is necessary. With the hybrid EA developed in this research, the CIN tree expands as the number of generations progresses. With the expansion of the genotype, the search space continues to expand, extending the time needed to develop a solution and posing a need for the ability to identify successful gene sets within the expanding data structure (Rosenman & Gero, 1999). The grafting method begins to address this as it captures the challenge of both identifying location (within the CIN) and the topology type in the selected cell patch. With the ability to graft this information into new locations, a new type of filter begins to emerge. In this proposition, the filter serves as a distribution tool and allows the genetic code to shrink with a subtree set of the CIN operating as a single gene. This modular approach has been referred to as artificial embryogenies (Manos et al., 2007). The manual intervention of grafting can serve to define new gene modules, returned back into the hybrid EA for continued development within the context of the entire mesh.

ACKNOWLEDGMENTS

The research was developed with the support of the Institute for Computational Design and the Institute for Building Structures and Structural Design at the University of Stuttgart and the Taubman College of Architecture and Urban Planning at the University of Michigan.

REFERENCES

Ahlquist, S., Kampowski, T., Oliyan Torghabehi, O., Menges, A., & Speck, T. (2014). Development of a digital framework for the computation of complex material and morphological behavior of biological and technological systems. *Computer-Aided Design 60*, 84–104.

Ahlquist, S., Lienhard, J., Knippers, J., & Menges, A. (2013). Physical and numerical prototyping for integrated bending and form-active textile hybrid structures. *Rethinking Prototyping: Proc. Design Modelling Symp., DMS '13.* Berlin: Springer.

Ahlquist, S., & Menges, A. (2010). Realizing formal and functional complexity for structurally dynamic systems in rapid computational means. *Proc. Advances in Architectural Geometry Conf., AGG '10.* Berlin: Springer.

Ahlquist, S., & Menges, A. (2013). Frameworks for computational design of textile micro-architectures and material behavior in forming complex force-active structures. *Adaptive Architecture Proc. Association for Computer Aided Design in Architecture, ACADIA '13*. Cambridge: Riverside Press.

Aish, R. (2011). DesignScript: origins, explanation, illustration. *Proc. Design Modeling Symp., DMS '11*, Berlin: Springer.

Aish, R. (2013). DesignScript: a learning environment for design computation. *Rethinking Prototyping: Proc. Design Modeling Symp., DMS '13.* Berlin: Springer.

Aish, R., Fisher, A., Joyce, S., & Marsh, A. (2012). Progress towards multi-criteria design optimisation using designscript with smart form, robot structural analysis and ecotect building performance analysis. *Synthetic Digital Ecologies, Proc. Association for Computer Aided Design in Architecture*, San Francisco, CA, October 18–21.

Baraff, D., & Witkin, A. (1998). Large steps in cloth simulation. *Proc. Computer Graphics and Interactive Techniques, SIGGRAPH '98.* Orlando, FL: ACM.

Bentley, P., & Corne, D. (2002). An introduction to creative evolutionary systems. In *Creative Evolutionary Systems* (Bentley, P.J., & Corne, D.W., Eds.), pp. 1–75. San Diego, CA: Academic Press.

Bentley, P., & Kumar, S. (1999). Three ways to grow designs: a comparison of embryogenies for an evolutionary design problem. *Proc. Genetic and Evolutionary Computation Conf., GECCO '99,* Orlando, FL, July 14–17.

Crutchfield, J.P. (1994). The calculi of emergence: computation, dynamics, and induction. *Physica D: Nonlinear Phenomena 75(1–3)*, 11–54.

Dianati, M., Song, I., & Treiber, M. (2002). *An introduction to genetic algorithms and evolution strategies.* Technical report, University of Waterloo.

Gerber, D. (2012). PARA—typing informing form and the making of difference. *International Journal of Architectural Computing 10(4)*, 501–520.

Greenwold, S. (2009). *Simong particles.* Accessed at http://www.bitbucket.org/simong/simong-particles on October 3, 2011.

Holland, J.H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology, Control and Artificial Intelligence.* Cambridge, MA: MIT Press.

Jones, G. (2002). Genetic and evolutionary algorithms. In *Encyclopedia of Computational Chemistry*. London: Wiley.

Kilian, A., & Ochsendorf, J. (2005). Particle-spring systems for structural form finding. *Journal of the International Association for Shell and Spatial Structures 46(148)*, 77–84.

Kim, I.Y., & de Weck, O.L. (2005). Variable chromosome length genetic algorithm for progressive refinement in topology optimization. *Structural and Multidisciplinary Optimization 29(6)*, 445–456.

Laumanns, M., Zitzler, E., & Thiele, L. (2000). A unified model for multi-objective evolutionary algorithms with elitism. *Proc. Congr. Evolutionary Computation*, La Jolla, CA, July 16–19, 2000.

Manos, S., Large, M.C.J., & Poladian, L. (2007). Evolutionary design of single-mode microstructured polymer optical fibres using an artificial embryogeny representation. *Proc. 9th Annual Conf. Genetic and Evolutionary Computation, GECCO '07*. London: ACM.

https://doi.org/10.1017/S0890060415000402 Published online by Cambridge University Press

Oliyan Torghabehi, O., & von Buelow, P. (2014). Performance oriented generative design of structural double skin facades inspired by cell morphologies. *Proc. Shells, Membranes and Spatial Structures, IASS-SLTE '14*, Brasilia, September 15–19.

O'Reilly, U.M., & Hemberg, M. (2007). Integrating generative growth and evolutionary computation for form exploration. *Genetic Programming and Evolvable Machines 8(2)*, 163–186.

Poli, R., Langdon, W., & McPhee, N. (2008). *A field guide to genetic programming*. Accessed at http://www.gp-field-guide.org.uk on June 15, 2014.

Rosenman, M.A., & Gero, J.S. (1999). Evolving designs by generating useful complex gene structures. In *Evolutionary Design by Computers* (Bentley, P., Ed.), pp. 345–364. San Francisco, CA: Morgan Kaufmann.

Russell, P.J. (1992). *Genetics*. New York: Harper Collins.

Ryoo, J., & Hajela, P. (2004). Handling variable string lengths in GA-based structural topology optimization. *Structural and Multidisciplinary Optimization 26(5)*, 318–325.

Schein, M., & Tessmann, O. (2008). Structural analysis as driver in surface-based design approaches. *International Journal of Architectural Computing 6(1)*, 19–39.

Schmidt, R. (2012). Interactive modeling with mesh surfaces. *Proc. ACM SIGGRAPH 2012*. Los Angeles, August 5–9, 2012.

Sedgewick, R., & Wayne, K. (2011). *Algorithms*. Boston: Pearson Education.

Shiue, L.-J., & Peters, J. (2005). Mesh refinement based on Euler encoding. *Proc. Shape Modeling and Applications SMI '05*, Cambridge, MA, June 13–17.

Terzopoulos, D., Platt, J., Barr, A., & Fleischer, K. (1987). Elastically deformable models. *Computer Graphics 21(4)*, 205–214.

Turner, J.S. (2007). *The Tinkerer's Accomplice*. Cambridge, MA: Harvard University Press.

von Buelow, P. (2007). *Genetically Engineered Architecture: Design Exploration With Evolutionary Computation*. Saarbrücken, Germany: AV Akademiker Verlag.

von Buelow, P. (2008). Breeding topology: special considerations for generative topology exploration using evolutionary computation. *Proc. Association for Computer Aided Design in Architecture Conf.*, pp. 346–353, Minneapolis, MN, October 16–19.

Welch, W., & Witkin, A. (1994). Free-form shape design using triangulated surfaces. *Proc. Conf. Computer Graphics and Interactive Techniques, SIGGRAPH '94*. New York: ACM.

Sean Ahlquist is an Assistant Professor at the University of Michigan Taubman College of Architecture and Urban Planning, where he develops research on material computation and codirects the MS in Material Systems Program. He is a part of the cluster in computational media and interactive systems that connects architecture with the fields of material science, computer science, art and design, and music. He is currently completing his doctorate (DrIng) at the University of Stuttgart Institute for Computational Design. His research is directed at formulating computational design frameworks that place materiality as an *a priori* agent in the organization of architectural systems and their spatial tectonics. In particular, the research explores technologies in the design and fabrication of variegated textile and fibrous material assemblages.

Dillon Erb received his Master of architecture degree from the University of Michigan Taubman College of Architecture and Urban Planning. His thesis explored computational alternatives to parametric modes of thinking in architecture. His current research focuses on the ways that design can benefit from advanced computational tool making, encompassing topics such as high-performance computing as a medium of design, and development of novel simulation methods and user interfaces.

Achim Menges is a registered Architect and Professor at the University of Stuttgart, where he is the Founding Director of the Institute for Computational Design. He is also currently a Visiting Professor of architecture at the Harvard University Graduate School of Design and Visiting Professor of emergent technologies and design graduate program at the Architectural Association, London. Prof. Menges's research and practice focuses on the development of integrative design processes at the intersection of morphogenetic design computation, biomimetic engineering, and digital fabrication. His projects and design research has received numerous international awards, and he has published and exhibited worldwide.

APPENDIX A

A.1. Performance notes on computational form-finding

The EA engine that was developed for springFORM is a multithreaded Java implementation capable of running a large number of individual form-finding simulations concurrently. Inherent limitations of the form-finding engine limit the effective number of particles and springs that any single simulation can quickly find an equilibrium within, but the multithreaded software is able to ameliorate this problem by running on as many processors/cores as the system has available. This system is "massively parallel" in the sense that performance directly (linearly) corresponds to the number of computer processors available. Tests are run on a workstation computer (2014 HP Z620) with 32 GB of RAM and a 3.2-GHz Intel Xeon CPU with 12 logical cores. The performance of the hybrid EA, where the largest determinant of the running time is the number of particles and springs within the system, is as follows:

150 pop size / 300 generations / ~2000 springs
= 45,000 individuals in 10 h 26 min

150 pop size / 289 generations / ~3000 springs
= 43,350 individuals in 22 h 36 min

No. of Particles/ Individual	No. of Individuals	Time to Solve	Avg. Time/ Individual
500	100	30 s	0.132 s
1000	100	37 s	0.37 s
2000	100	246 s	2.46 s
4000	100	383 s	3.83 s
500	200	75.29 s	0.376 s
1000	200	70 s	0.376 s
2000	200	594 s	2.97 s
3000	200	900 s	4.5 s

https://doi.org/10.1017/S0890060415000402 Published online by Cambridge University Press