Nature-Based Hybrid Computational Geometry System for Optimizing Component Structure

Danil Nagy(✉), Dale Zhao, and David Benjamin

The Living, an Autodesk Studio, New York, NY, USA
danil.nagy@autodesk.com

Abstract. This paper describes a novel computational geometry system developed for application in the design of full-scale industrial components. This system combines a bottom-up growth strategy based on slime mould behaviour in nature with a top-down genetic algorithm strategy for optimization. The growth strategy uses an agent-based algorithm to create individual instances of designs based on a small number of input parameters. These parameters can then be controlled by a genetic algorithm to optimize the final design according to goals such as minimizing weight and minimizing structural weakness. Together, these two strategies create a hybrid approach which ensures high performance while allowing the designer to explore a wider range of novel designs than would be possible using traditional design methods.

Keywords: Design and modelling of matter · Multi-objective optimization · Generative design · Computational geometry · Additive manufacturing

Introduction

The Design Problem

The hybrid computational geometry system described in this paper was developed in partnership with a team of researchers at a large aircraft manufacturer and applied to the redesign of a partition inside a commercial aircraft (Fig. 1). The partition is the wall that divides the seating area from the galley, and the goal for the project was to reduce its weight by 50%. This weight reduction is critical to the aerospace industry to reduce fuel consumption, cost of flying, and carbon emissions.

While the partition wall may seem like a relatively simple component, it actually presents two complex structural challenges. First, the partition must support a fold-down cabin attendant seat (CAS). Unlike the partition, the CAS is not attached to the airplane's fuselage or the floor, thus the full weight of two flight attendants and the seat itself must be transferred through the partition into the aircraft's structure. Since the CAS is hanging from the partition, this creates an asymmetrical load. And to pass certification, the partition must withstand a crash test in which the weight of the CAS and its attendants is accelerated to 16 times the force of gravity (16G)—an extremely challenging structural task.

K. De Rycke et al., *Humanizing Digital Reality*,
https://doi.org/10.1007/978-981-10-6611-5_15

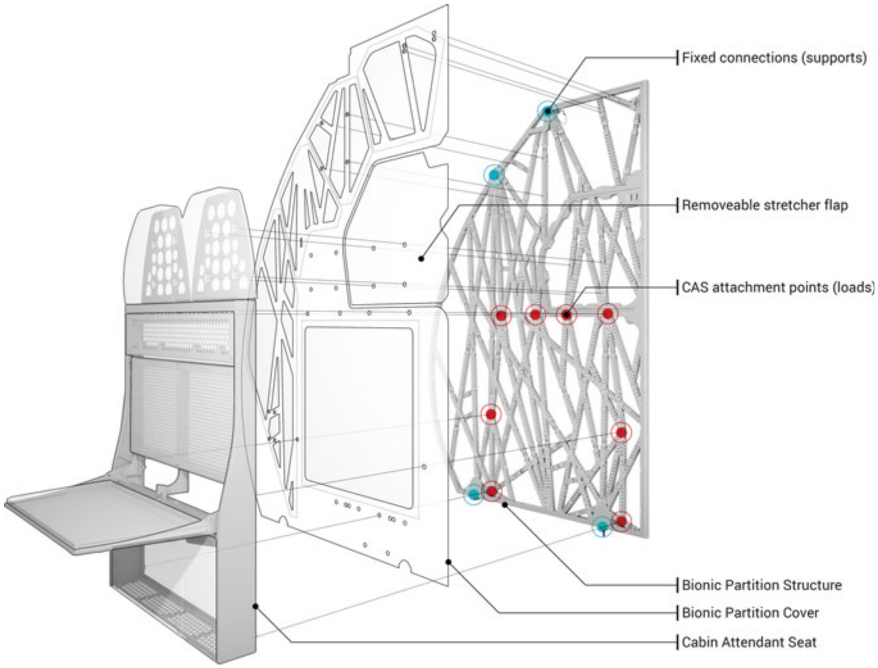

Fig. 1. Description of aircraft cabin partition design problem

Second, due to new safety regulations, the partition must include a panel called the ‘stretcher flap’ which can be removed to allow a stretcher carrying a sick or injured passenger to be carried around the corner from the seating area to the galley and exit. This results in a big hole in the partition which makes it difficult to route forces from the CAS directly into the aircraft’s fuselage.

Due to these structural challenges, the state-of-the-art partition design is very heavy and expensive. The goal of our collaboration was to develop a computational design workflow based on natural intelligence that could leverage the potential of metal additive manufacturing to create the next generation of lightweight, strong, and affordable aircraft components.

Designing for Complexity

New technologies of metal additive manufacturing have made possible the fabrication of fully usable industrial components with complex geometries that could not be manufactured using traditional methods. Although additive manufacturing processes come with their own set of limitations in terms of what can be produced and the human labor needed to produce it, in general printing the same volume of material costs the same regardless of its formal complexity. This contrasts with traditional methods such as machining or casting, where formal complexity is often a significant aspect of the part’s final cost.

These new capabilities in manufacturing have opened new design possibilities that have only barely been explored. One issue is that high-resolution formal complexity is very difficult to comprehend using traditional tools and design methods. While 3D printers can easily describe surface features to the tenth of a millimeter, there are no existing design tools which would allow the human designer to reason or design at this level of detail. To take advantage of the opportunities of these new manufacturing methods, we need new computational design tools that can assist us in the exploration of this huge space of potential designs and find the best performing solutions to our design problems.

Evolving Design

A common tool in the exploration of complex, highly multidimensional design spaces is the genetic algorithm (GA). The GA is a particularly popular example of a meta-heuristic search algorithm (Yang and Luniver Press 2010) which can explore a 'black box' parametric model to find the highest performing designs based on one or more objectives. Unlike other optimization methods based on gradient descent, GAs are completely top-down and do not require any knowledge of the model in order to mine it for the best results. This is well-suited for optimizing design models which are typically defined by a large set of geometric operations, all of which may be difficult or impossible to analytically describe or differentiate.

To utilize a GA to solve a design problem, the designer must specify a generative geometry system that describes a 'design space' of possible solutions to the problem, as well as one or more measurable goals for the design. In an engineering context, performance goals tend to be fairly straight forward, and usually involve maximizing structural performance (by minimizing stress, displacement, etc.) while minimizing the amount of material used. Thus, our main challenge in this project was to develop a unique generative geometry system which could create a wide set of possible designs in a way that could be optimized by a GA. This paper describes the system we developed, demonstrates how it was used to redesign the aircraft partition, and speculates about potential applications of such systems for future design problems.

Literature Review

The use of genetic algorithms for structural optimization is well explored in the literature, and Marler and Arora (2004) provide a good overview of the subject. This problem is usually formulated as the optimization of the topology, shape, and size (TSS) of a structural layout given a set of loading conditions and an arrangement of supports. While structural optimization is a broad field, a more specific subtopic that relates to this paper focuses on the way in which structures are represented and how they are parametrized so that they can be explored productively by a GA—in other words, how the design system's input parameters, or genotype, relates to the physical form, or phenotype, of each design iteration. Aulig and Olhofer (2016) provide a good overview of the variety of representation strategies that have been developed and break down the strategies into three basic categories: (1) grid, (2) geometric, and (3) indirect.

The first two representational strategies use a direct parameterization of geometry. In the grid strategy, the entire space of the potential structure is discretized into an orthogonal grid whose values represent either the presence of material (binary encoding) or its density (real-valued encoding) at each location. In the geometric strategy, the structure is represented directly as a collection of geometric shapes. For example, a truss structure may be represented by a collection of straight-line beams which encode the size and shape of each element.

The indirect strategy, on the other hand, uses an intermediate system to generate the structure. Two common techniques for such indirect parameterization are rule-based algorithms such as Lindenmeyer systems (Hornby and Pollack 2001) and behavioural algorithms such as cellular automata (Mitchell et al. 1996). With this approach, parameters are used to control either the rules or behaviours of the intermediate algorithm, which is then executed to create the actual structure.

Although they are more complex, indirect strategies offer several advantages over direct representations, including easier scalability to address larger design problems (Kicinger et al. 2004). In the context of design, they also offer the benefit of using a relatively small number of input parameters to define a highly complex design space with a wide variety of design solutions that may not be intuitive to the designer. Combined with an optimization process based on the GA, such complex design spaces are more likely to produce design solutions that are not only high performing but also novel and unexpected.

Because of the indirect nature of this kind of parameterization, there is a huge potential for designers to invent new types of indirect representations which are customized to address specific design problems. This paper contributes to this endeavour by describing a novel indirect representation based on a behavioural algorithm inspired by the growth of slime mould in nature.

Methodology

The following section describes the details of our generative geometry system, including the parameterized behavioural algorithm which creates each design iteration, the set of measures which determine the performance of each design, and the specification of the optimization process based on the genetic algorithm which was used to achieve the final design.

Design Model

The geometry of each partition design is created by a behavioural 'bottom-up' algorithm which is inspired by the growth of slime mould in nature. As slime mould grows, it first spreads out a dense network of connections. Then, based on where it finds food, it starts to prune the network to keep only those connections that most efficiently connect the food sources. This behaviour forms complex adaptive networks that are efficient and redundant. They are efficient because they use a small amount of "lines" to connect a set of "points" (sources of food). They are redundant because when one of the lines is damaged, the network can often "route around" the problem and keep the

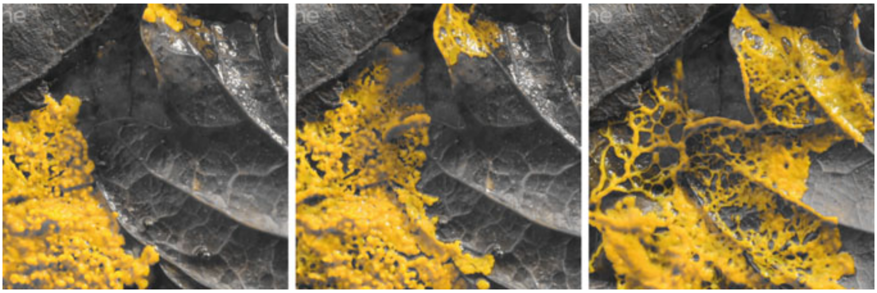

Fig. 2. Growth of slime mould over time showing initial dense network and later pruning to the most important connections

points connected (Tero et al. 2010). Overall, the living slime mould organism produces complex, highly-efficient food distribution networks based only on local behaviour (Fig. 2).

Although the geometry of slime mould growth is based on connecting points (food sources) in 2D to transport nutrients, we hypothesized that a similar logic may be beneficial for connecting points (structural attachment points) in 2D to resist structural loads. To enact a slime mould geometry system in our design model, we first specified a set of ‘seed points’ which consisted of the 4 support points (where the partition attaches to the fuselage), the 16 load points (where the CAS attaches to the partition), and 28 additional points sampled evenly along the partition boundary. Then, a ground structure (Bendsøe et al. 1994) of all valid structural connections is defined as the set of all straight-line segments between two seed points which do not cross the boundary of the panel. This ground structure is encoded as a graph whose vertices are the seed points and whose edges are the structural elements.

Next, a real-valued weight parameter in the domain [0, 1] is assigned to each vertex of the graph. The structural members of each design iteration are then sampled from the edges of the graph based on the following algorithm:

For each structural member (s):

1. Locate the vertex with the highest weight (w)
2. Select the edge which connects this vertex with its highest weight (w) neighbor
3. Decay the weight of both vertices by multiplying it by a decay parameter (d)

The final structural design is defined by the boundary of the panel plus the set of all selected edges (Fig. 3). Because all load points need to be connected into the structure for the design to be valid, a final step checks each load point to see if it was connected during the main growth step. If not, an additional structural member is created from the point to the closest point on the structure.

Like the slime mould, our model starts with a dense network of possible connections. The weights assigned to each seed point represent a varying quantity of food at each point, and structural pathways are selected for the design based on those that connect the highest food quantities. Just as the slime mould eats the food causing its

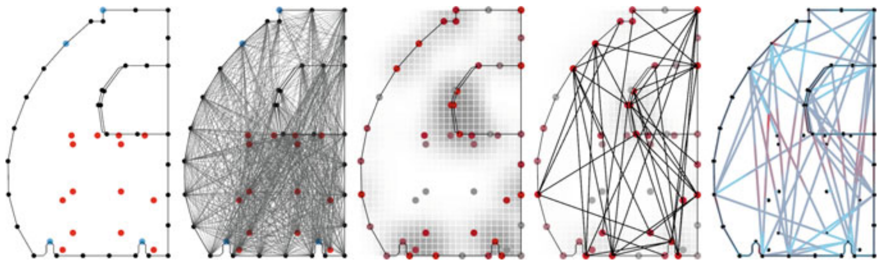

Fig. 3. Diagram of computational geometry system based on growth of slime mould

network to evolve over time, the decay factor slowly reduces the weight of each utilized seed point allowing connections to grow in other parts of the structure.

The parameters of this model are the weights (w) of the 48 seed points, plus the number of structural members (s) and the decay parameter (d). Since all the parameters are continuous, the GA is able to "learn" how to work with the growth behaviour and tune it to create better performing designs over time.

Model Evaluation

This behavioural generative geometry model can create a large variety of structural designs for the partition based on a relatively small set of input parameters. However, in order to use a genetic algorithm to evolve high-performing designs, the model must also contain a set of measures which tell the algorithm which designs are better performing. Our model uses static finite element analysis (FEA) to simulate the performance of each design under the given loading conditions. This analysis gives us a set of metrics which we can use to establish the objectives and constraints of our optimization problem:

1. Total partition weight. This should be minimized (objective).
2. Maximum displacement, which is how much the panel moves under loading. This should be less than 2 mm based on the given performance requirements (constraint)
3. Maximum utilization, which is the percentage of the maximum stress allowance of the material experienced by the structural members. This should be less than 50% based on a standard safety factor (constraint).

In addition to these structural goals and constraints, we specified an additional design objective to maximize the distribution of material (minimize the number of large holes) within the perimeter of the partition. This is to discourage designs which solve the structural loading problem while leaving large holes in the structure which may cause other problems when passengers or objects bump into the partition. This set of two objectives and two constraints completes the specification of the model (Fig. 4) and allows the genetic algorithm to automatically search the range of possible designs to find a set of valid and optimized designs.

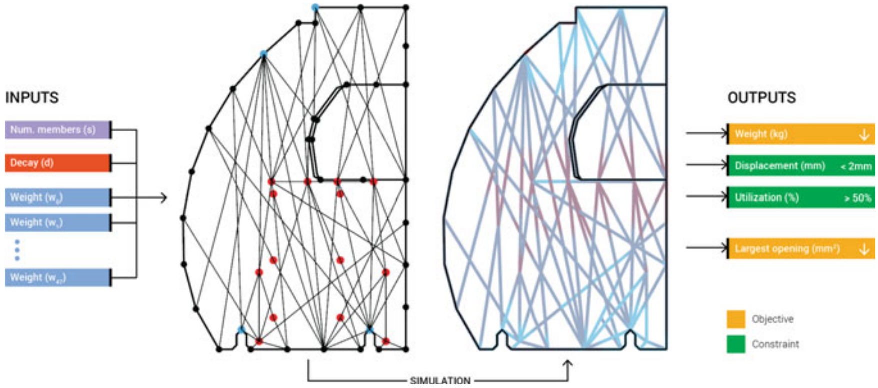

Fig. 4. Diagram of geometry system with 50 inputs, 2 constraints derived from FEA simulation, and two objectives derived from the geometry of the model

Model Optimization

Using this model, we performed an optimization using a variant of the NSGA-II genetic algorithm (Deb et al. 2002) with the following settings:

- Number of designs per generation: 200
- Number of generations: 100
- Mutation rate: 0.05
- Cross-over rate: 0.9

Once the optimization is complete, we can visualize the results by plotting them relative to the objectives of the optimization. Figure 6 shows each partition design explored by the optimization plotted as a point on a scatter plot where the x-axis represents the weight of the partition, the y-axis represents the infill factor, and the colour represents the generation in which it was created. The squares are the invalid designs based on the two constraints.

The goal of the optimization is to find designs which meet the two structural constraints with a minimum weight and a minimum number of large holes. Because these two objectives are in competition with each other, there is no single best solution (Fig. 5). Thus, the goal of the genetic algorithm is to discover the set of 'Pareto optimal' solutions that each solve the trade-off between these competing objectives in a different way. Figure 6 shows the ability of the genetic algorithm to develop subsequent generations of designs which push the 'boundary' of optimal designs closer and closer to the conceptual point of optimal performance in the lower left hand corner of the plot. The set of Pareto optimal designs are shown with a thick black outline in Fig. 6, and a subset of them is visualized above the figure.

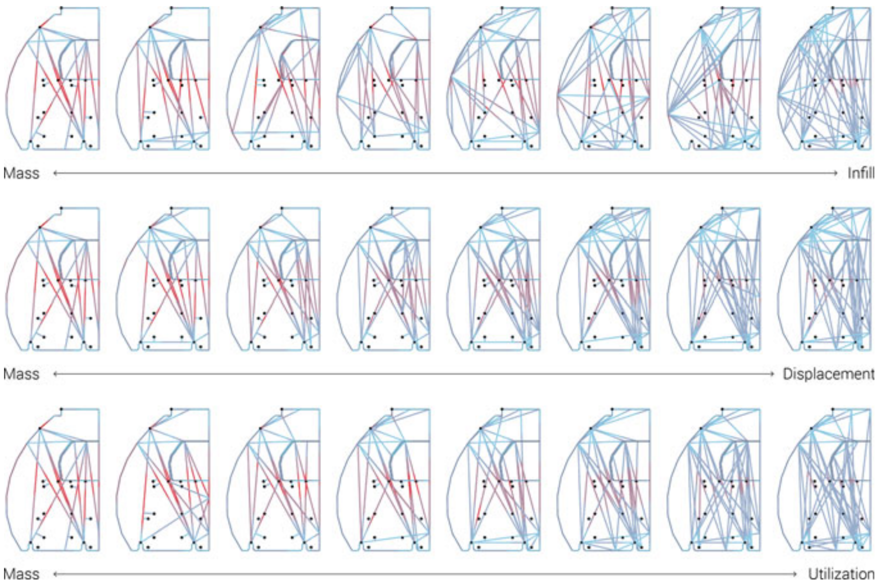

Fig. 5. Sample of designs showing trade-offs between sets of two objectives

After Optimization

Once the optimization process was complete, we selected the final design as the one which minimally met our 50% weight reduction goal while creating the least large holes in the structure (second from right in Fig. 6). The design was then further developed by breaking each of the structural members optimized through the evolutionary process into a set of smaller lattice structures. Each of these lattice beams was further optimized by changing its diameter based on the local stress distribution in the structure. This secondary optimization allowed us to fine tune the component structure so that we exactly met the performance requirement while reducing the weight as much as possible. Finally, in order to manufacture the component we needed to break it down into a set of smaller components which could fit into the bed of a metal selective laser sintering (SLS) machine (Fig. 7).

Once the parts were manufactured we assembled them into the final partition shown in Fig. 7. This part met our design goals: having the same structural performance as the state-of-the-art component with only 50% of the weight. The partition created through

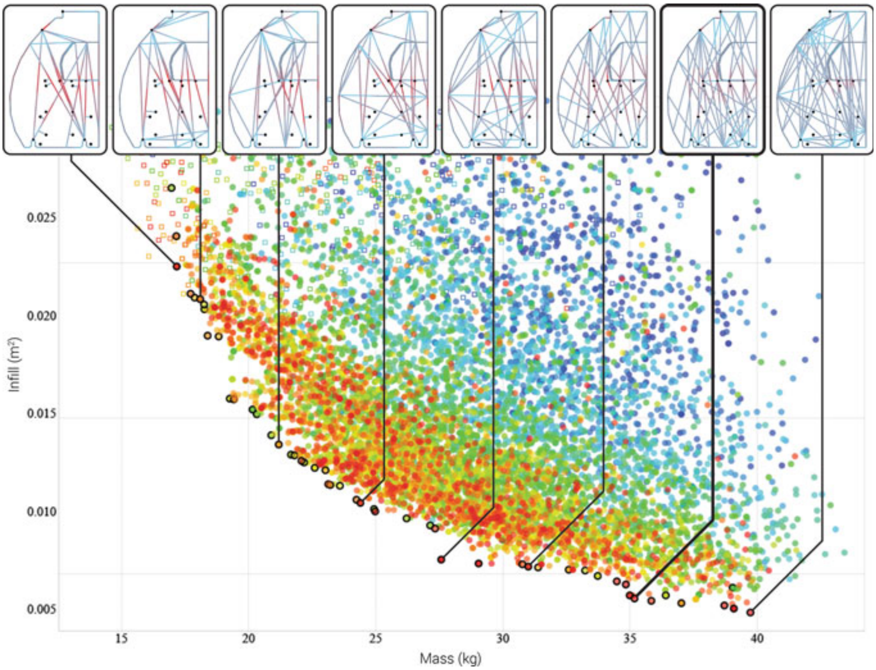

Fig. 6. All designs explored during the optimization process plotted according to the two objectives. *Colour* represents the generation in which the design was evaluated, with *blue* for earlier and *red* for later designs. Designs with a *black outline* are part of the Pareto-dominant set of optimal solutions

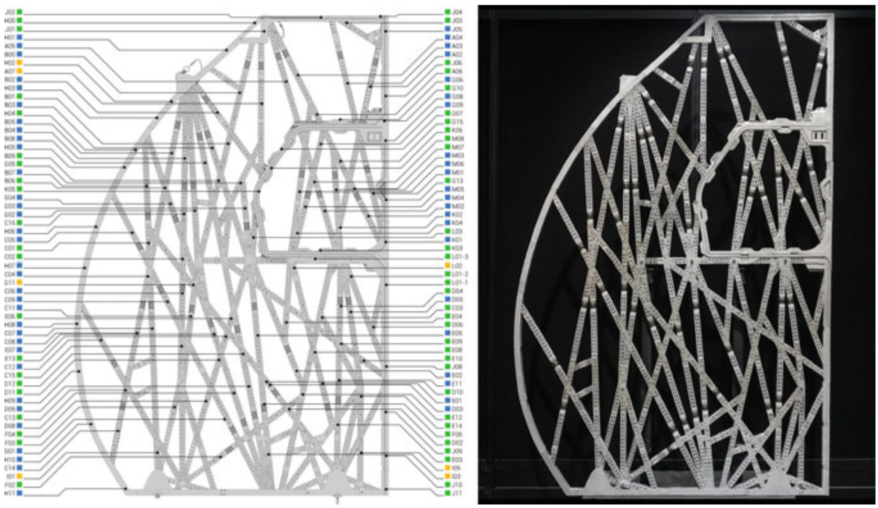

Fig. 7. Images of final design including diagram of component breakdown (*left*) and photograph of printed partition prototype (*right*)

this process is the largest aircraft component ever produced entirely through metal additive manufacturing, and is currently undergoing testing and certification which will allow it to be integrated into future airplanes for commercial flights.

Conclusions

This paper describes a novel computational design method which combines a generative geometry model based on the 'bottom-up' agent-based growth processes found in natural systems with a 'top-down' genetic algorithm for optimization. The paper also describes the application of this method toward the design of a unique industrial component which can take advantage of the formal freedoms allowed by recent advances in metal additive manufacturing. Finally, our method suggests future research into the development of other nature-based generative design systems which can leverage the power of evolutionary computing to derive unique, high-performing solutions to complex design challenges.

References

Aulig, N., Olhofer, M.: Evolutionary computation for topology optimization of mechanical structures: an overview of representations. In: 2016 IEEE Congress on Evolutionary Computation (CEC), IEEE, pp. 1948–1955 (2016)

Bendsøe, M.P., Ben-Tal, A., Zowe, J.: Optimization methods for truss geometry and topology design. Struct. Multidiscip. Optim. **7**(3), 141–159 (1994)

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.A.M.T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)

Hornby, G.S., Pollack, J.B.: Evolving L-systems to generate virtual creatures. Comput. Graph. **25** (6), 1041–1048 (2001)

Kicinger, R., Arciszewski, T., De Jong, K.: Morphogenesis and structural design: cellular automata representations of steel structures in tall buildings. In: Congress on Evolutionary Computation, 2004. CEC2004, vol. 1, pp. 411–418. IEEE (2004)

Marler, R.T., Arora, J.S.: Survey of multi-objective optimization methods for engineering. Struct. Multidiscip. Optim. **26**(6), 369–395 (2004)

Mitchell, M., Crutchfield, J.P., Das, R.: Evolving cellular automata with genetic algorithms: a review of recent work. In: Proceedings of the First International Conference on Evolutionary Computation and Its Applications (EvCA'96) (1996)

Tero, A. et al.: Rules for biologically inspired adaptive network design. Science (2010)

Yang, X.-S.: Nature-Inspired Metaheuristic Algorithms. Luniver Press (2010)