Lecture

Constraints & Optimization

Amit Zoran

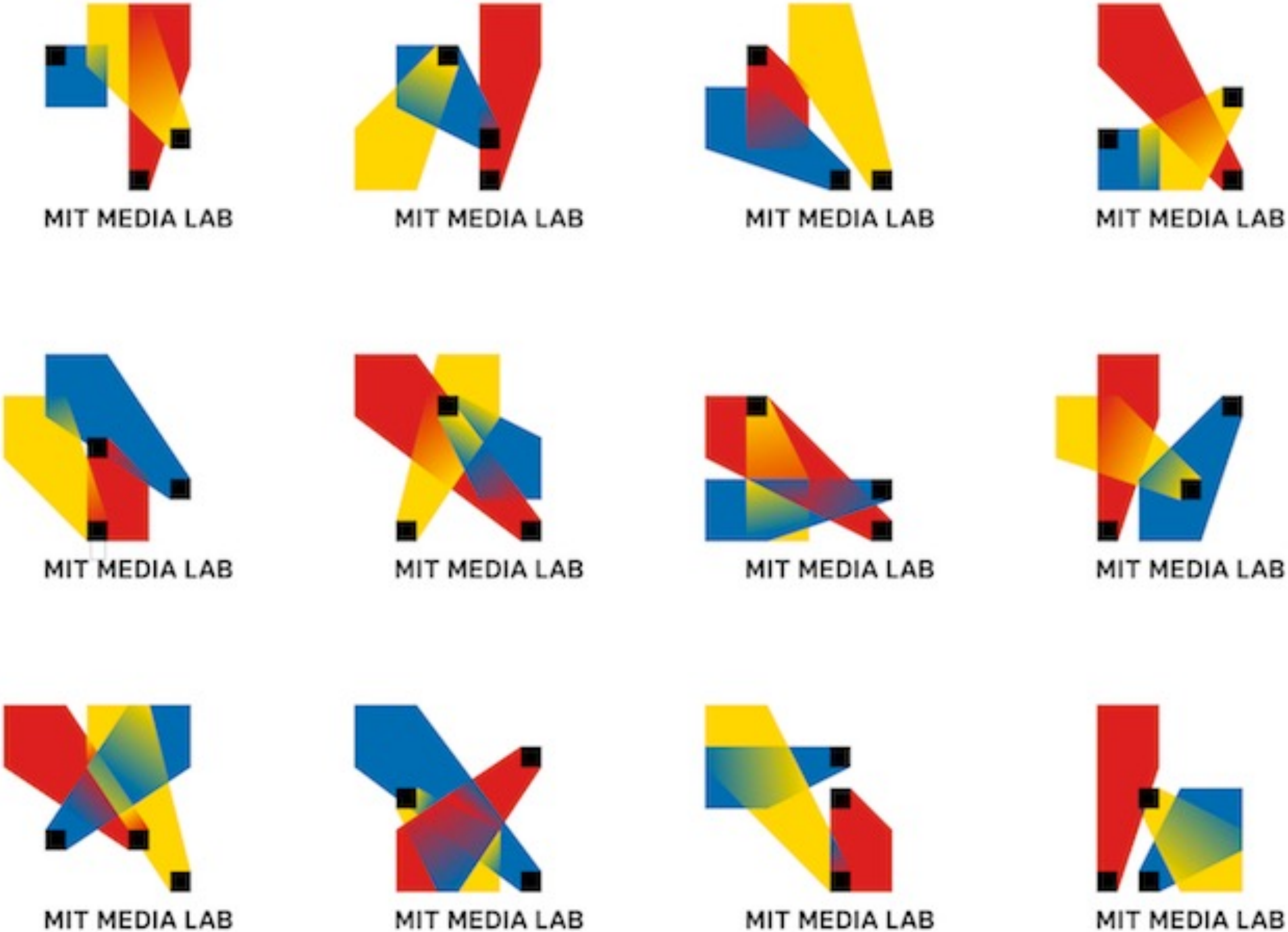
Advanced Topics in Digital Design

67682

The Rachel and Selim Benin School of Computer Science and Engineering
The Hebrew University of Jerusalem, Israel

Outline

- The Parametric Design Space
- Introduction to Optimization
- Introduction to Gradient Descent
- Grasshopper Galapagos
- Example for Genetic Algorithms in Parametric Design



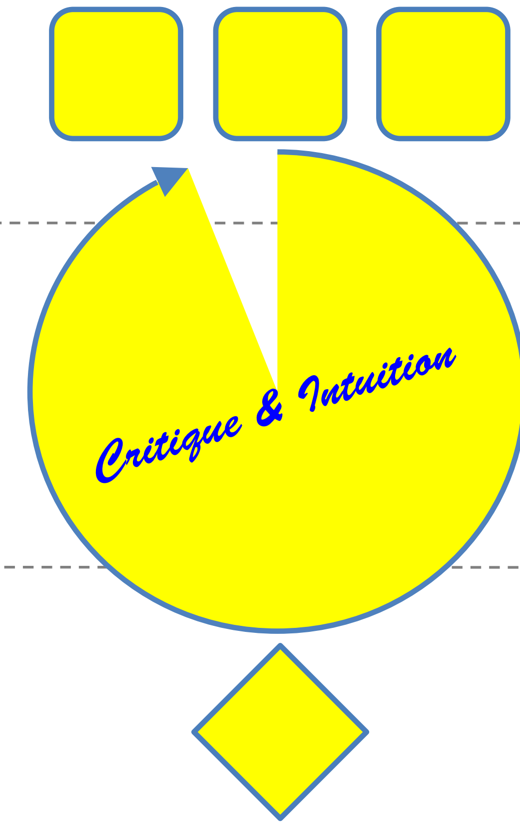
By E Roon Kang and Richard The

https://vimeo.com/20250134

The design process: critique, intuition, and skills

Requirements & constraints

- Prioritizing



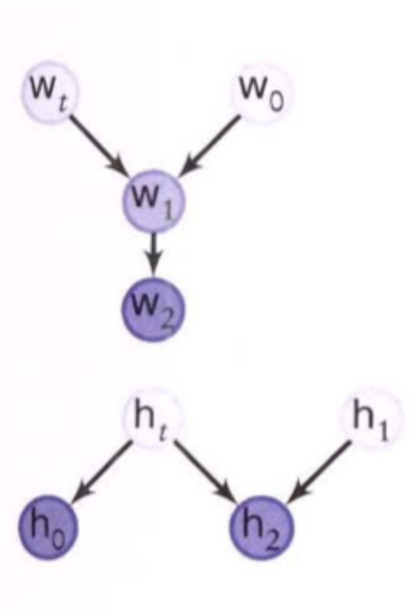
The modeling process

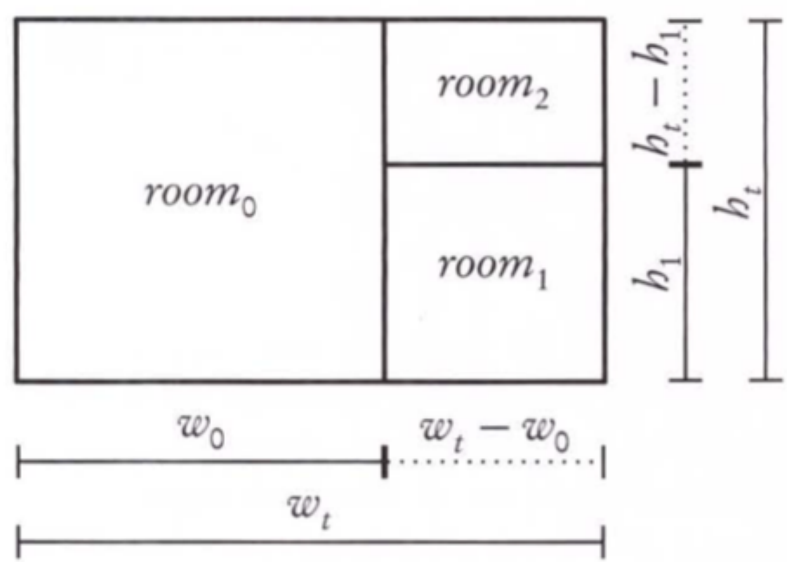
- Search
- Implementation
- Evaluation

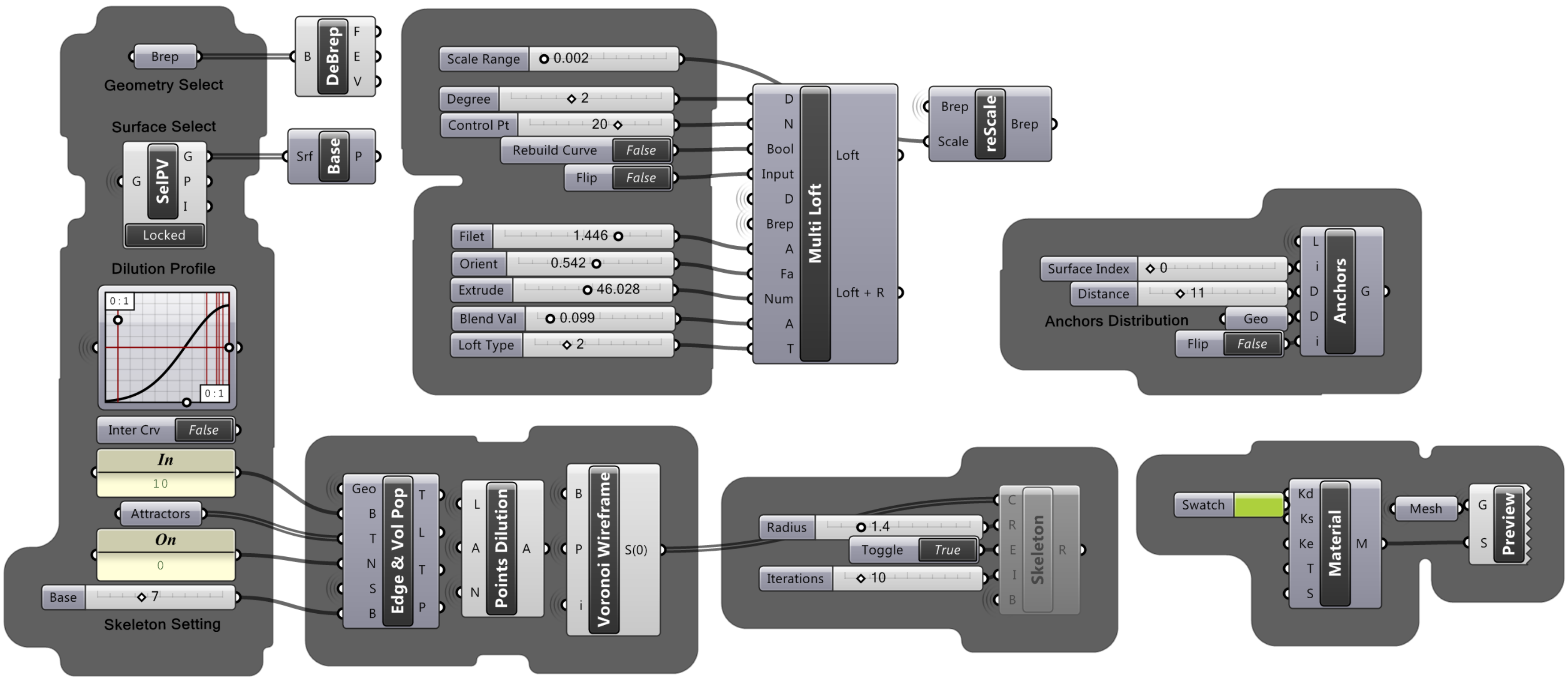
The output product

- Mockups & prototypes
- Rendering
- Working model / product

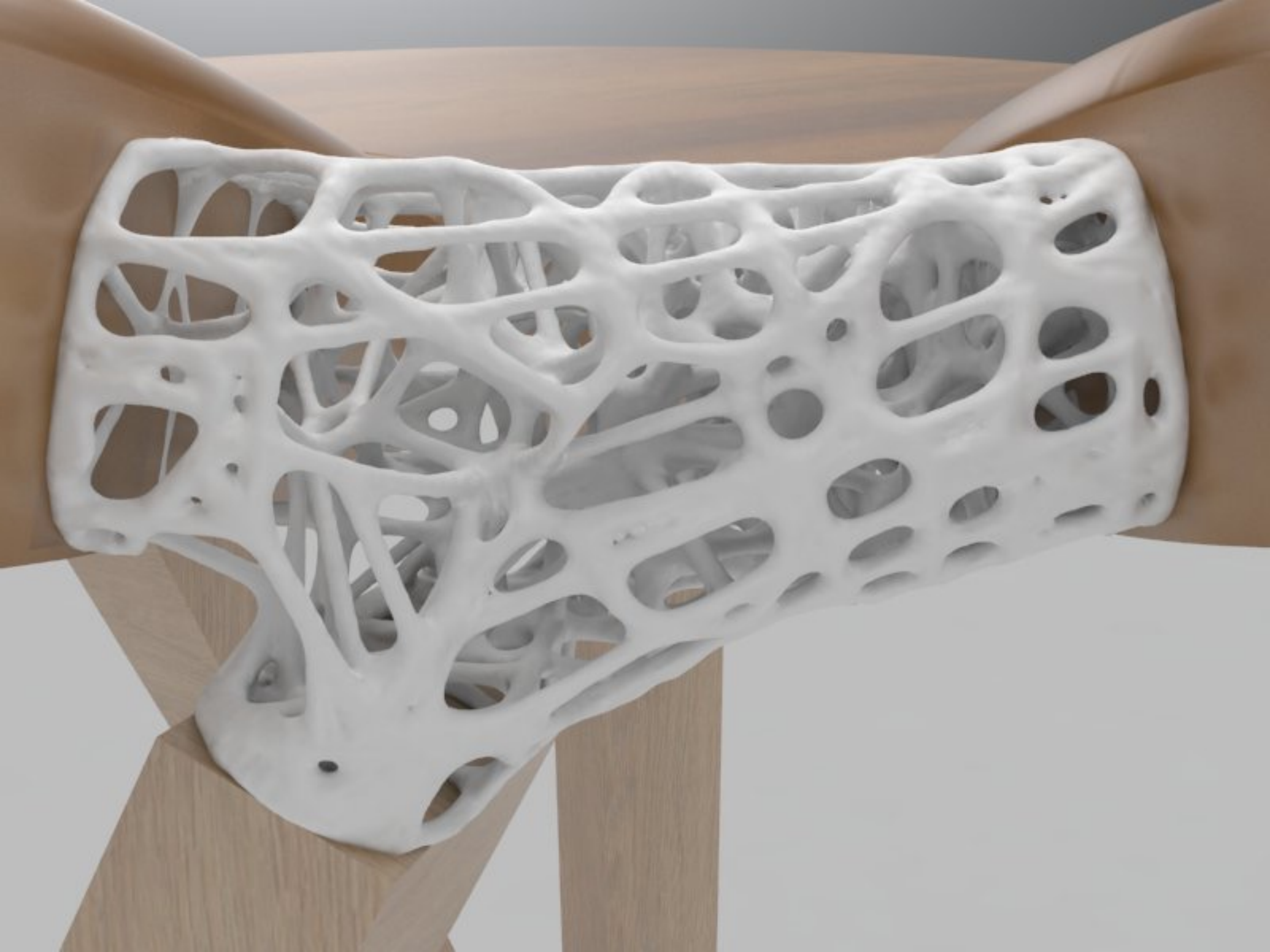
The Parametric Design Space

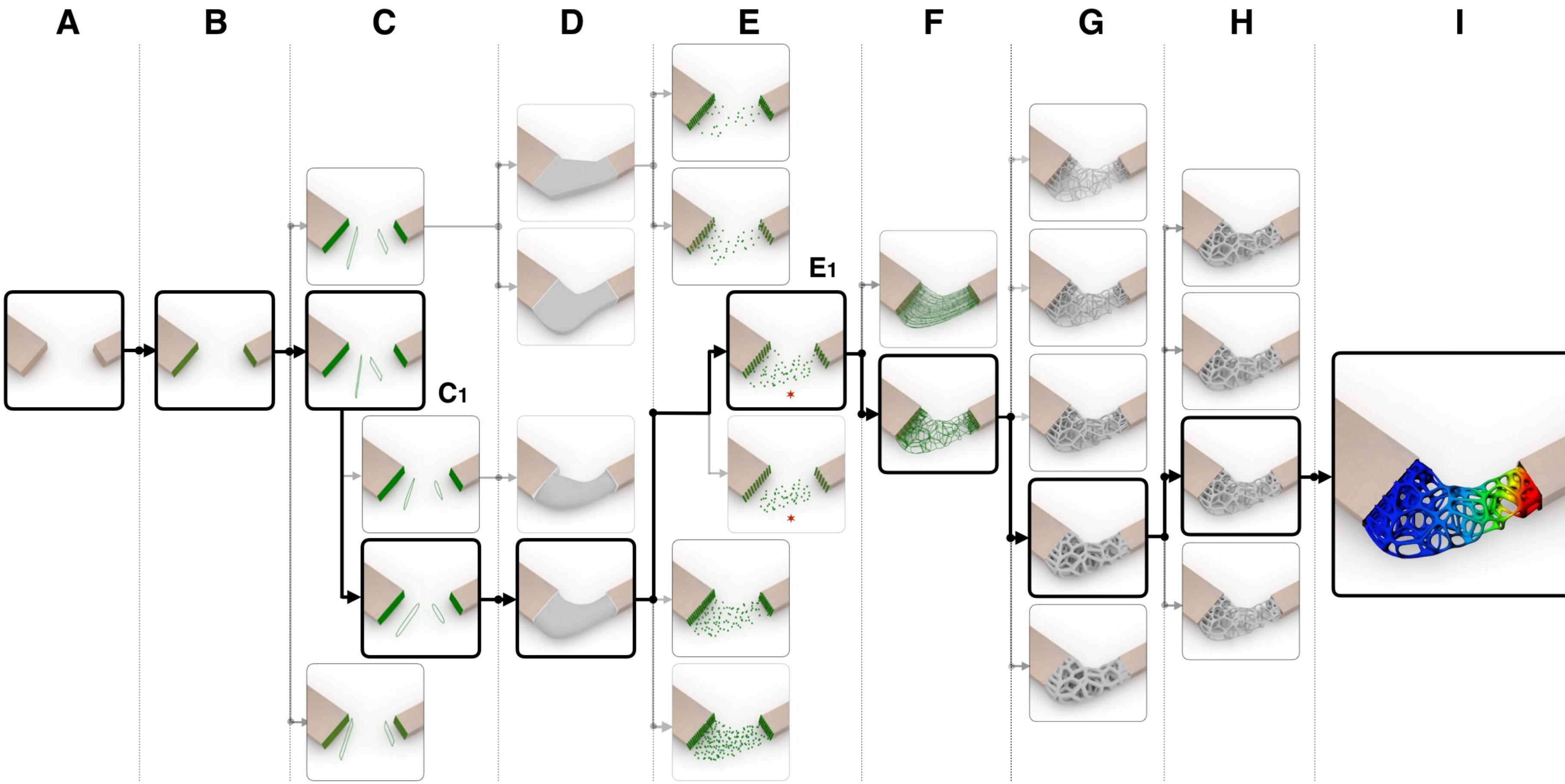


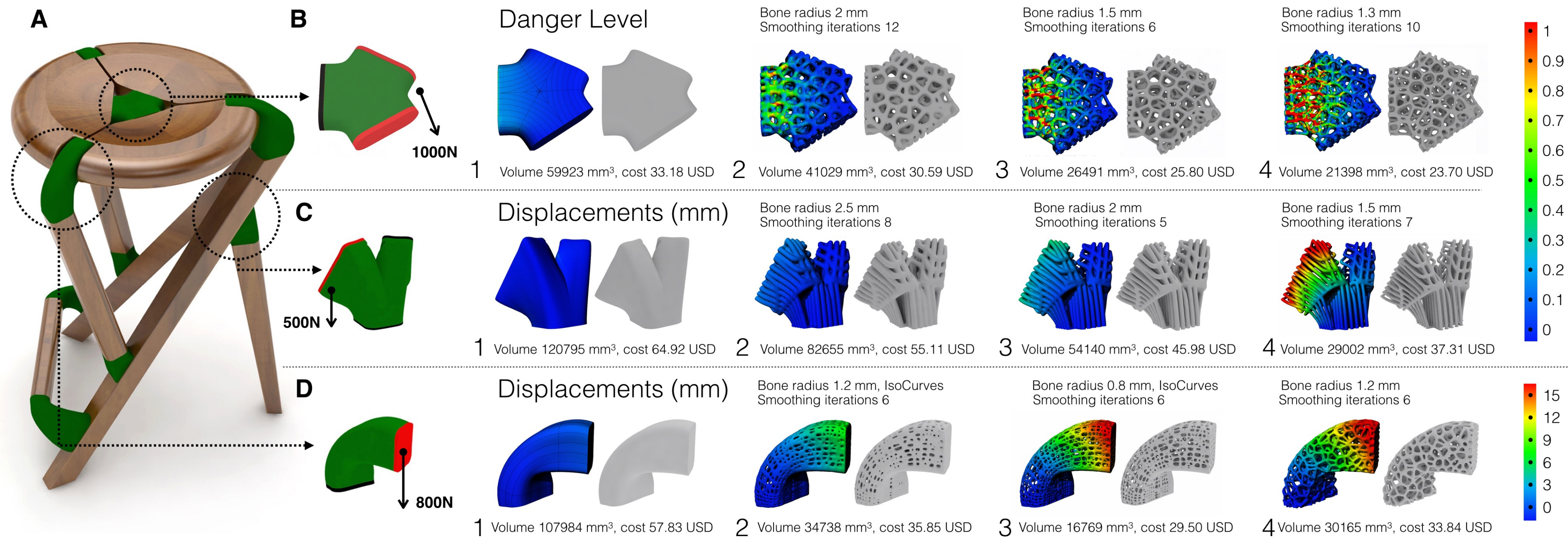




Brep
Geometry Select
DeBrep
Surface Select
SelPV
Locked
Srf
Base
Dilution Profile
Inter Crv
False
In
10
Attractors
On
0
Base
7
Skeleton Setting
Scale Range
0.002
Degree
2
Control Pt
20
Rebuild Curve
False
Flip
False
Filet
1.446
Orient
0.542
Extrude
46.028
Blend Val
0.099
Loft Type
2
Multi Loft
Loft
Loft + R
reScale
Brep
Scale
Surface Index
0
Distance
11
Anchors Distribution
Geo
Flip
False
Anchors
Edge & Vol Pop
Points Dilution
Voronoi Wireframe
S(0)
Radius
1.4
Toggle
True
Iterations
10
Skeleton
Swatch
Material
Mesh
Preview




A
B
C
D
E
F
G
H
I
C1
E1


A
B
1000N
Danger Level
1 Volume 59923 mm³, cost 33.18 USD
Bone radius 2 mm
Smoothing iterations 12
2 Volume 41029 mm³, cost 30.59 USD
Bone radius 1.5 mm
Smoothing iterations 6
3 Volume 26491 mm³, cost 25.80 USD
Bone radius 1.3 mm
Smoothing iterations 10
4 Volume 21398 mm³, cost 23.70 USD
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
C
500N
Displacements (mm)
1 Volume 120795 mm³, cost 64.92 USD
Bone radius 2.5 mm
Smoothing iterations 8
2 Volume 82655 mm³, cost 55.11 USD
Bone radius 2 mm
Smoothing iterations 5
3 Volume 54140 mm³, cost 45.98 USD
Bone radius 1.5 mm
Smoothing iterations 7
4 Volume 29002 mm³, cost 37.31 USD
D
800N
Displacements (mm)
1 Volume 107984 mm³, cost 57.83 USD
Bone radius 1.2 mm, IsoCurves
Smoothing iterations 6
2 Volume 34738 mm³, cost 35.85 USD
Bone radius 0.8 mm, IsoCurves
Smoothing iterations 6
3 Volume 16769 mm³, cost 29.50 USD
Bone radius 1.2 mm
Smoothing iterations 6
4 Volume 30165 mm³, cost 33.84 USD
15
12
9
6
3
0

Introduction to Optimization Problems

In mathematics and computer science, an optimization problem is the problem of finding the best solution from all feasible solutions. (Wikipedia)

Types of Optimization Problems

- Continuous Optimization Vs. Discrete Optimization
- Unconstrained Optimization versus Constrained Optimization
- None, One or Many Objectives
- Deterministic Optimization versus Stochastic Optimization

From https://neos-guide.org/optimization-tree

Maximum / Minimum Problems

Problems of the first derivative

Guidelines for solving Max./Min Problems

1. Read each problem slowly and carefully. Read the problem at least three times before trying to solve it. It is imperative to know exactly what the problem is asking. If you misread the problem or hurry through it, you have NO chance of solving it correctly.

2. If appropriate, draw a sketch or diagram of the problem to be solved. Pictures are a great help in organizing and sorting out your thoughts.

3. Define variables to be used and carefully label your picture or diagram with these variables. This step is very important because it leads directly or indirectly to the creation of mathematical equations.

From https://www.math.ucdavis.edu/~kouba/CalcOneDIRECTORY/maxmindirectory/MaxMin.html

Maximum / Minimum Problems

Problems of the first derivative

Guidelines for solving Max./Min Problems (continue)

4. Write down all equations which are related to your problem or diagram. Clearly denote that equation which you are asked to maximize or minimize. MOST optimization problems will begin with two equations. One equation is a "constraint" equation and the other is the "optimization" equation. The "constraint" equation is used to solve for one of the variables. This is then substituted into the "optimization" equation before differentiation occurs. Some problems may have NO constraint equation. Some problems may have two or more constraint equations.

5. Before differentiating, make sure that the optimization equation is a function of only one variable. Then differentiate using the well-known rules of differentiation.

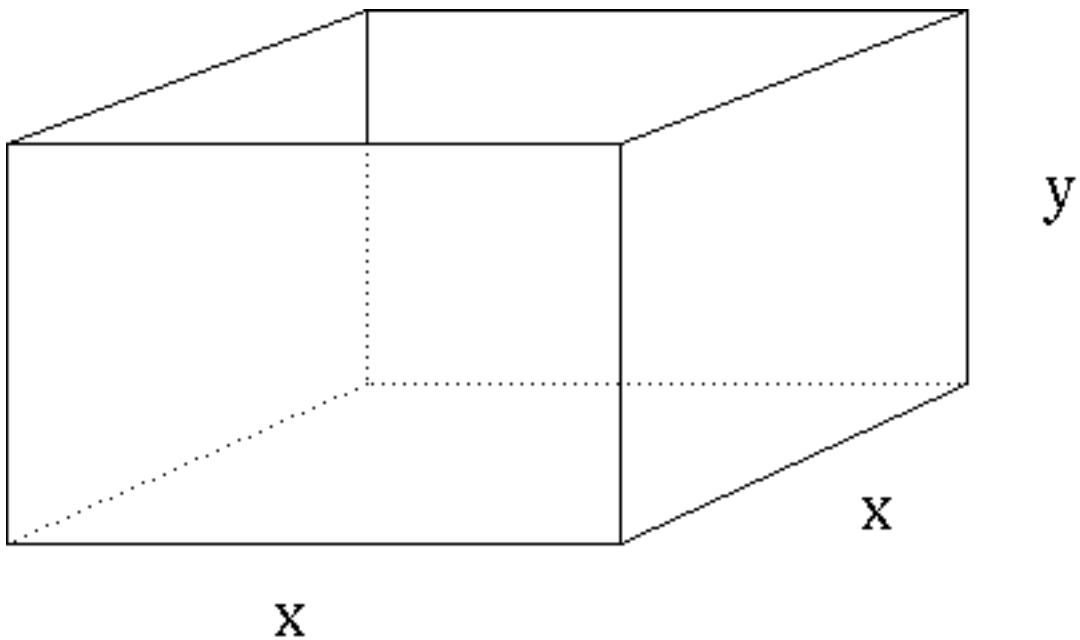
6. Verify that your result is a maximum or minimum value using the first or second derivative test for extrema.

From https://www.math.ucdavis.edu/~kouba/CalcOneDIRECTORY/maxmindirectory/MaxMin.html

Maximum / Minimum Problems

PROBLEM (EXAMPLE 1): An open rectangular box with square base is to be made from 48 ft.^{2} of material. What dimensions will result in a box with the largest possible volume ?

:



From https://www.math.ucdavis.edu/~kouba/CalcOneDIRECTORY/maxmindirectory/MaxMin.html

Maximum / Minimum Problems

SOLUTION: Let variable x be the length of one edge of the square base and variable y the height of the box. The total surface area of the box is given to be

48 = (area of base) + 4 (area of one side) = $x^2 + 4(xy)$,

so that $4xy = 48 - x^2$

$$y = \frac{48 - x^2}{4x}$$

$$= \frac{48}{4x} - \frac{x^2}{4x}$$

$$= \frac{12}{x} - (1/4)x \ .$$

From https://www.math.ucdavis.edu/~kouba/CalcOneDIRECTORY/maxmindirectory/MaxMin.html

Maximum / Minimum Problems

We wish to MAXIMIZE the total VOLUME of the box

V = (length) (width) (height) = (x) (x) (y) = x^2y .

However, before we differentiate the right-hand side, we will write it as a function of x only. Substitute for y getting $V = x^2 y = 12x - (1/4)x^3$

Now differentiate this equation, getting

$V' = 12 - (1/4)3x^2 = 12 - (3/4)x^2 = (3/4)(16 - x^2) = (3/4)(4 - x)(4 + x) = 0$

for

x=4 or x=-4 .

But x > 0 . Since the base of the box is square and there are 48 ft^2 of material, it follows that $0 < x < 48^{0.5}$.

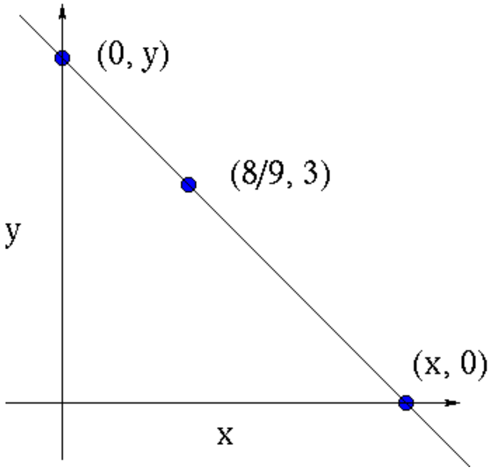
So, If x=4 ft. and y=2ft., then V = 32 $ft.^3$

From https://www.math.ucdavis.edu/~kouba/CalcOneDIRECTORY/maxmindirectory/MaxMin.html

Maximum / Minimum Problems

PROBLEM (EXAMPLE 2): Consider all triangles formed by lines passing through the point (8/9, 3) and both the x- and y-axes. Find the dimensions of the triangle with the shortest hypotenuse.

SOLUTION: Let variable x be the x-intercept and variable y the y-intercept of the line passing through the point (8/9, 3) .



$$\frac{y}{x} = \frac{3}{x - 8/9}$$

$$y = \frac{3x}{x - 8/9}$$

From https://www.math.ucdavis.edu/~kouba/CalcOneDIRECTORY/maxmindirectory/MaxMin.html

Maximum / Minimum Problems

We wish to MINIMIZE the length of the HYPOTENUSE of the triangle

$$H = \sqrt{x^2 + y^2}$$

However, before we differentiate the right-hand side, we will write it as a function of x only. Substitute for y getting

$$H = \sqrt{x^2 + y^2} = \sqrt{x^2 + \left(\frac{3x}{x - 8/9}\right)^2}$$

Now differentiate this equation using the chain rule and quotient rule, getting

$$H' = (1/2)\left(x^2 + \left(\frac{3x}{x - 8/9}\right)^2\right)^{-1/2} \left\{2x + 2\left(\frac{3x}{x - 8/9}\right)\frac{(x - 8/9)(3) - (3x)(1)}{(x - 8/9)^2}\right\}$$

$$= (1/2)\left(x^2 + \left(\frac{3x}{x - 8/9}\right)^2\right)^{-1/2} (2)\left\{x + \frac{3x}{(x - 8/9)}\frac{-8/3}{(x - 8/9)^2}\right\} = \frac{x - \dfrac{8x}{(x - 8/9)^3}}{\sqrt{x^2 + \left(\dfrac{3x}{x - 8/9}\right)^2}} = 0$$

Maximum / Minimum Problems

so that (If A/B = 0, then A=0)

$$x - \frac{8x}{(x-8/9)^3} = 0$$

$$x\left\{1 - \frac{8}{(x-8/9)^3}\right\} = 0$$

so that (If AB= 0 , then A=0 or B=0) x=0

(Impossible, since x> 8/9. Why ?) or

$$1 - \frac{8}{(x-8/9)^3} = 0$$

$$1 = \frac{8}{(x-8/9)^3}$$

x = 26/9 and y=13/3, $H = \frac{13\sqrt{13}}{9} \approx 5.21$

Gradient Descent

Gradient descent is a first-order iterative optimization algorithm.

To find a local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient (or of the approximate gradient) of the function at the current point. If instead one takes steps proportional to the positive of the gradient, one approaches a local maximum of that function; the procedure is then known as gradient ascent. Gradient descent is also known as steepest descent, or the method of steepest descent.

Wikipedia

Gradient Descent

Gradient descent is based on the observation that if the multi-variable function $F(x)$ is defined and differentiable in a neighborhood of a point a, then $F(x)$ decreases fastest if one goes from a in the direction of the negative gradient of F at a, $-\nabla F(a)$.

It follows that, if

$$\mathbf{a^{n+1}} = \mathbf{a^n} - \gamma \nabla F(\mathbf{a^n})$$

for γ small enough, then $F(a^n) >= F(a^{n+1})$.

In other words, the term $\gamma \nabla F(a)$ is subtracted from a because we want to move against the gradient, namely down toward the minimum.

Gradient Descent

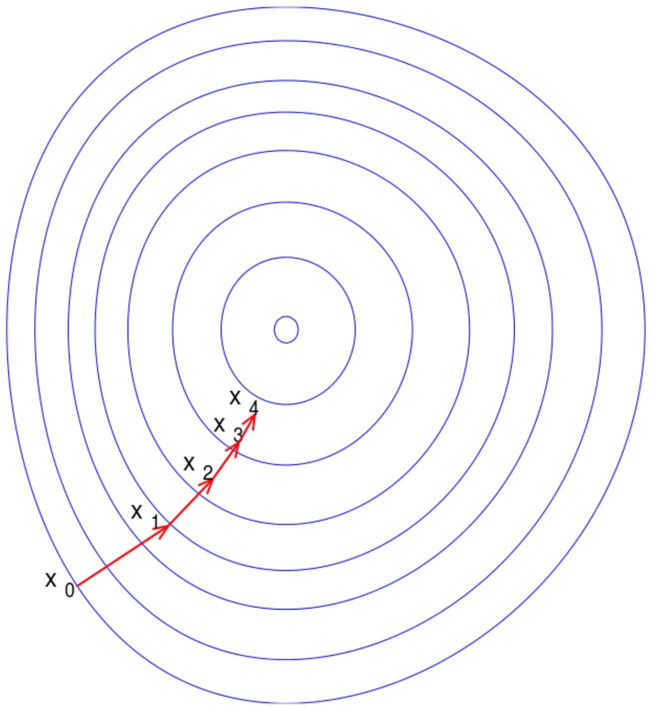
With this observation in mind, one starts with a guess x_0 for a local minimum of F, and considers the sequence x_0, x_1, x_2 such that

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \gamma_n \nabla F(\mathbf{x}_n),\ n \geq 0.$$

we have

$$F(\mathbf{x}_0) \geq F(\mathbf{x}_1) \geq F(\mathbf{x}_2) \geq \cdots,$$

so hopefully the sequence x_n converges to the desired local minimum.



Grasshopper Galapagos - Genetic Algorithm

Evolutionary Principles applied to Problem Solving

Cons

Evolutionary Algorithms are (very) slow.

Evolutionary Algorithms do not guarantee a solution.

Pros

Evolutionary Algorithms are remarkably flexible.

Evolutionary Algorithms are also quite forgiving.

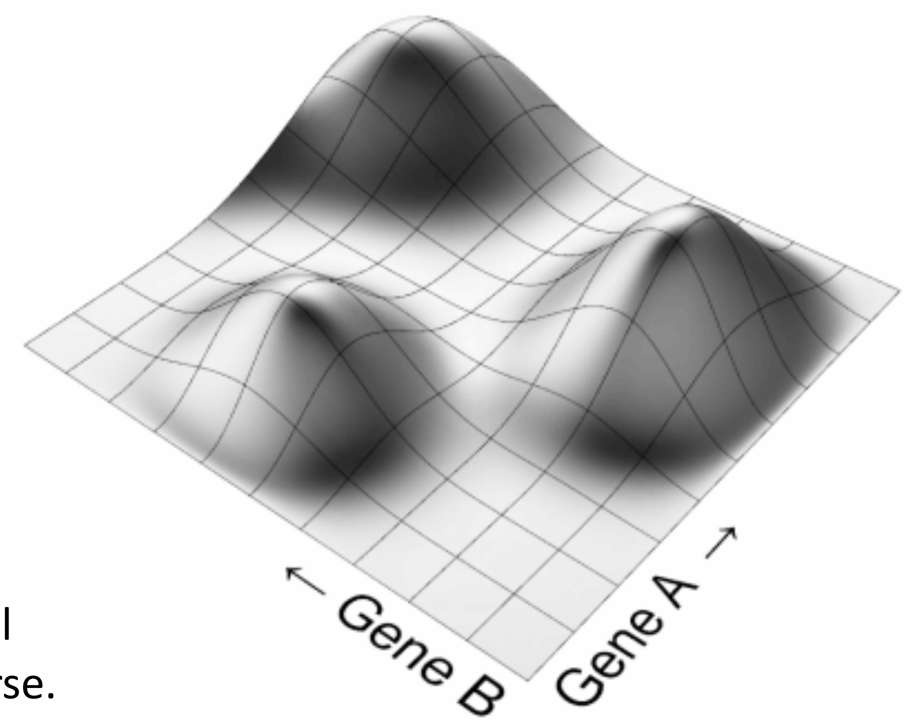
Evolutionary Solvers allow for a high degree of interaction with the user.

From David Rutten's blog http://www.grasshopper3d.com/profiles/blogs/evolutionary-principles

Grasshopper Galapagos - Genetic Algorithm

Evolutionary Principles applied to Problem Solving

The Process: Example with a *Fitness Landscape of a particular model*



The model contains two variables (genes).

As we change Gene A, the state of the model changes and it either becomes better or worse.

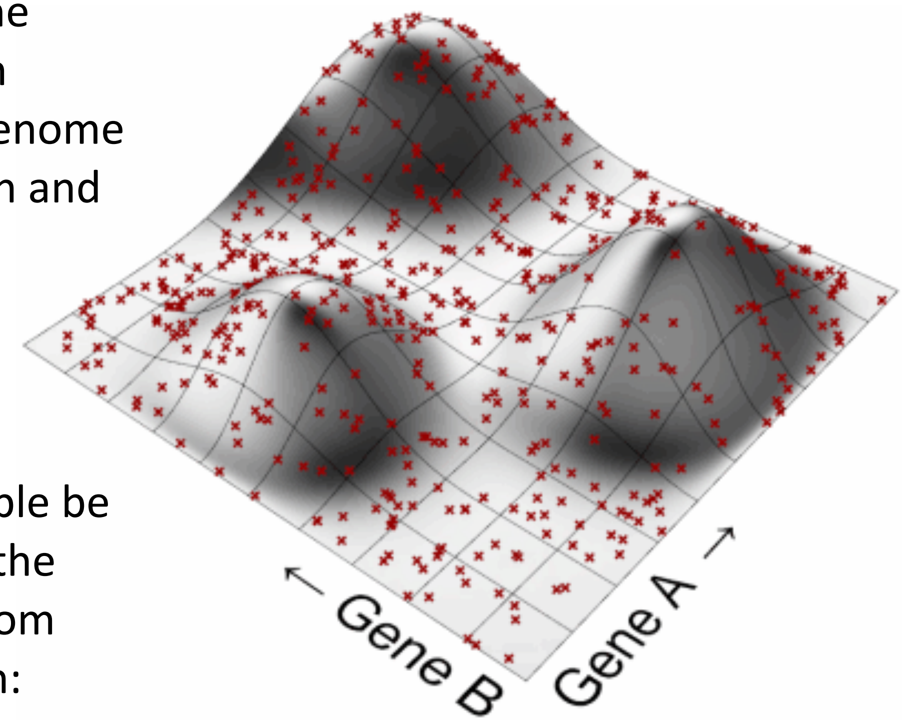
From David Rutten's blog http://www.grasshopper3d.com/profiles/blogs/evolutionary-principles

Grasshopper Galapagos - Genetic Algorithm

Evolutionary Principles applied to Problem Solving

The Process: Example with a *Fitness Landscape of a particular model*

The initial step of the solver is to populate the landscape (or "model-space") with a random collection of individuals (or "genomes"). A genome is nothing more than a specific value for each and every gene.



In the above case, a genome could for example be {A=0.2 B=0.5}. The solver will then evaluate the fitness for each and every one of these random genomes, giving us the following distribution:

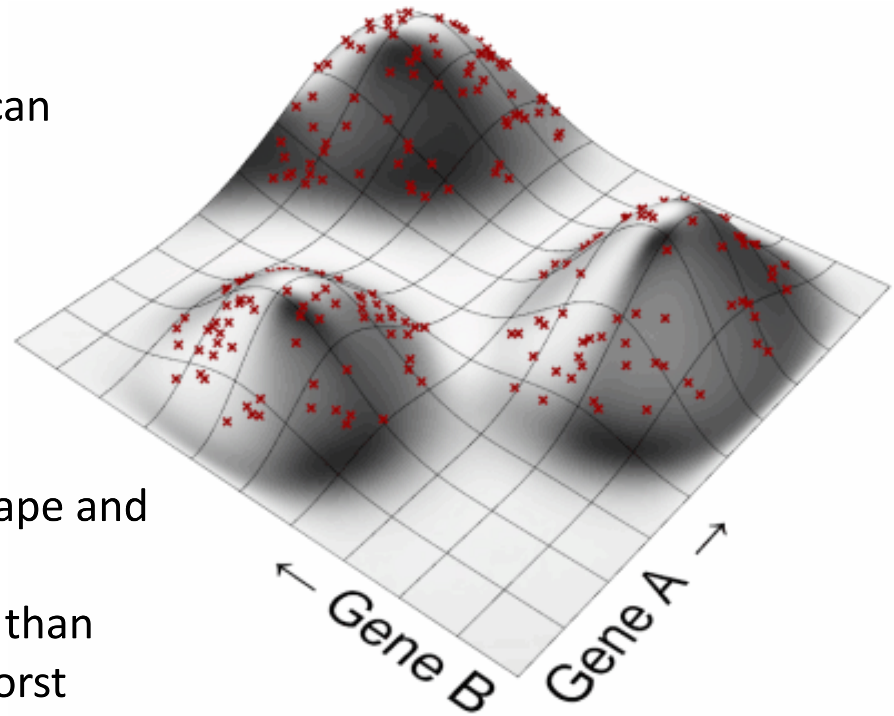
From David Rutten's blog http://www.grasshopper3d.com/profiles/blogs/evolutionary-principles

Grasshopper Galapagos - Genetic Algorithm

Evolutionary Principles applied to Problem Solving

The Process: Example with a *Fitness Landscape of a particular model*

Once we know how fit every genome is, we can make a hierarchy from fittest to lamest.



We are looking for high-ground in the landscape and it is a reasonable assumption that the higher genomes are closer to potential high-ground than the low ones. Therefore we can kill off the worst performing ones and focus on the remainder.

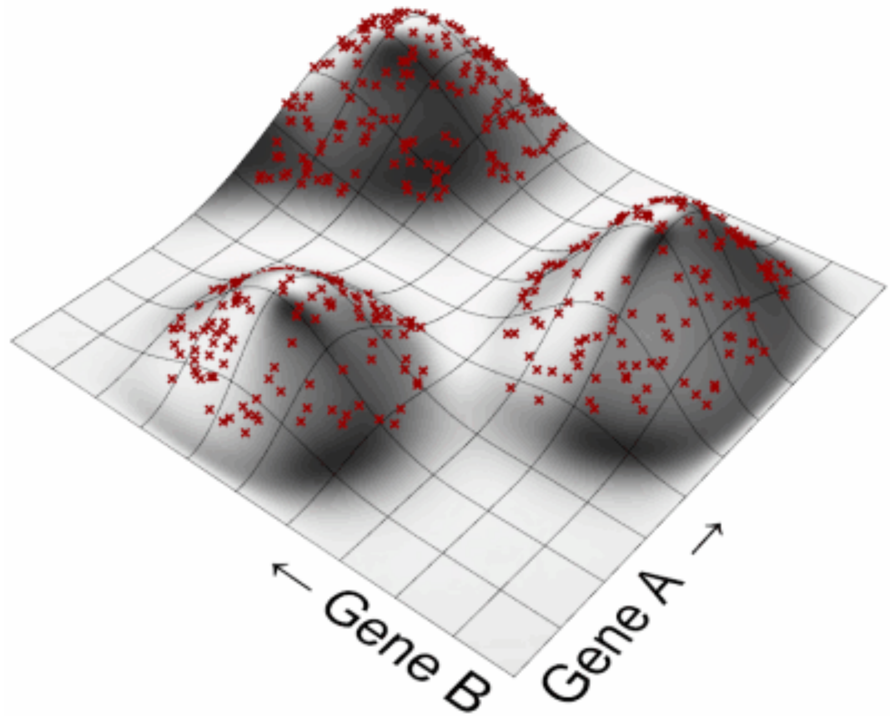
From David Rutten's blog http://www.grasshopper3d.com/profiles/blogs/evolutionary-principles

Grasshopper Galapagos - Genetic Algorithm

Evolutionary Principles applied to Problem Solving

The Process: Example with a *Fitness Landscape of a particular model*

Breeding the best performing genomes in Generation 0 to create Generation 1



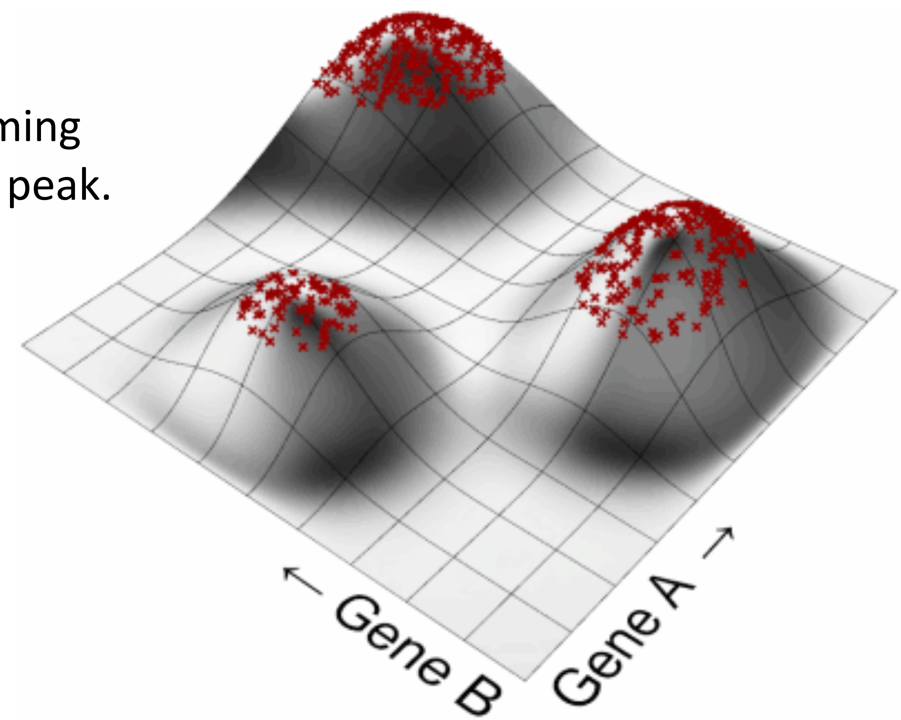
From David Rutten's blog http://www.grasshopper3d.com/profiles/blogs/evolutionary-principles

Grasshopper Galapagos - Genetic Algorithm

Evolutionary Principles applied to Problem Solving

The Process: Example with a *Fitness Landscape of a particular model*

Repeating the above steps (kill off the worst performing genomes, breed the best-performing genomes) until we have reached the highest peak.



From David Rutten's blog http://www.grasshopper3d.com/profiles/blogs/evolutionary-principles

Grasshopper Galapagos - Genetic Algorithm

Evolutionary Principles applied to Problem Solving

In order to perform this process, an Evolutionary Solver requires five interlocking parts, We could call this the anatomy of the Solver.

- Fitness Function
- Selection Mechanism
- Coupling Algorithm
- Coalescence Algorithm
- Mutation Factory

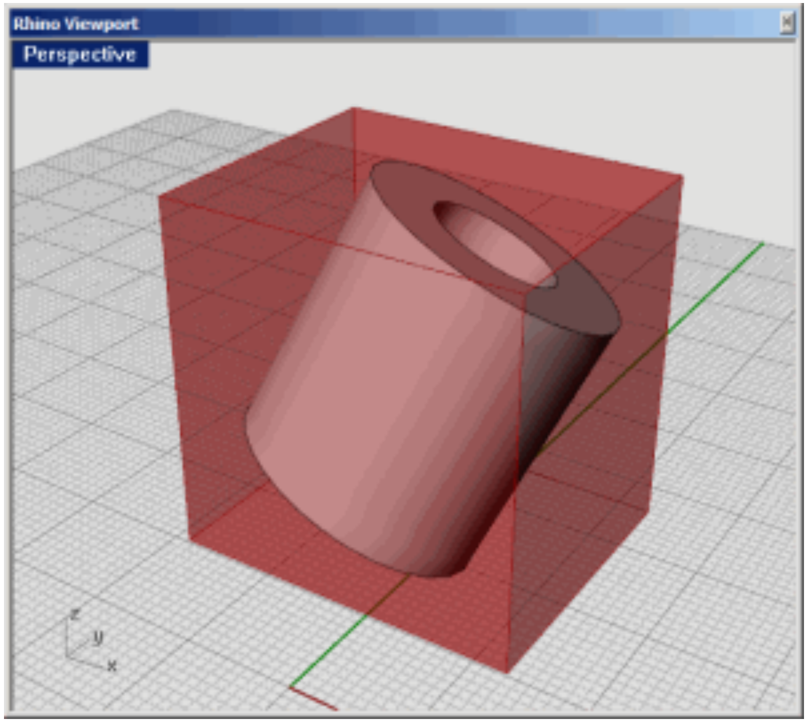
From David Rutten's blog http://www.grasshopper3d.com/profiles/blogs/evolutionary-principles

Grasshopper Galapagos - Evolutionary Principles applied to Problem Solving

Fitness Functions

Fitness is whatever we want it to be. We are trying to solve a specific problem, and therefore we know what it means to be fit. If for example we are seeking to position a shape so that it may be milled with minimum material waste, there is a very strict fitness function that leaves no room for argument.

Let's imagine the fitness landscape represents a model that seeks to encase an object in a minimum volume bounding-box. A minimum bounding-box is the smallest orthogonal box that completely contains any given shape.



From David Rutten’s blog http://www.grasshopper3d.com/profiles/blogs/evolutionary-principles

Grasshopper Galapagos - Evolutionary Principles applied to Problem Solving

Selection Mechanisms (that available in Galapagos)

Isotropic Selection, which is the simplest kind of algorithm you can imagine. It dampens the speed with which a population runs uphill. It therefore acts as a safe-guard against a premature colonization of a local -and possibly inferior- optimum.

Another mechanism available in Galapagos is **Exclusive Selection**, where only the top N% of the population get to mate. If you're lucky enough to be in the top N%, you'll likely have multiple offspring.

Another common pattern in nature is **Biased Selection**, where the chance of mating increases as the fitness increases. Biased Selection can be amplified by using power functions, which have the effect of flattening or exaggerating the curve.

From David Rutten’s blog http://www.grasshopper3d.com/profiles/blogs/evolutionary-principles

Grasshopper Galapagos - Evolutionary Principles applied to Problem Solving

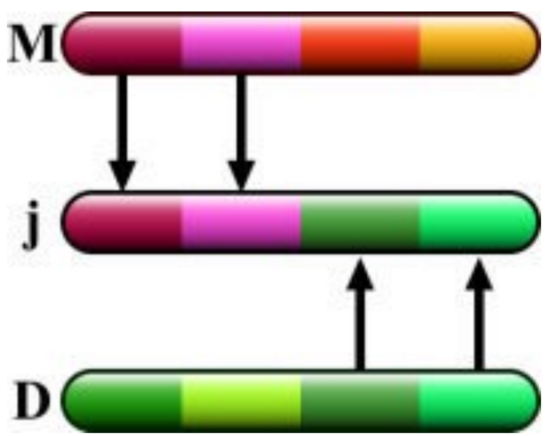
Coupling Algorithms

Coupling is the process of finding mates. Once a genome has been elected to mate by the active Selection Algorithm, it has to pick a mate from the population to complete the act. There are of course many ways in which mate selection could occur, but Galapagos at the moment only allows one; selection by genomic distance.

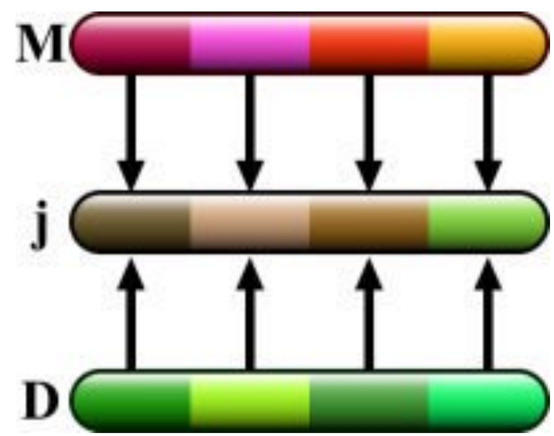
Grasshopper Galapagos - Evolutionary Principles applied to Problem Solving

Coalescence Algorithms

Once a mate has been selected, offspring needs to be generated. Genes in evolutionary solvers like Galapagos behave like floating point numbers, that can assume all the values between two numerical extremes.



Crossover Coalescence



Blend Coalescence

From David Rutten's blog http://www.grasshopper3d.com/profiles/blogs/evolutionary-principles

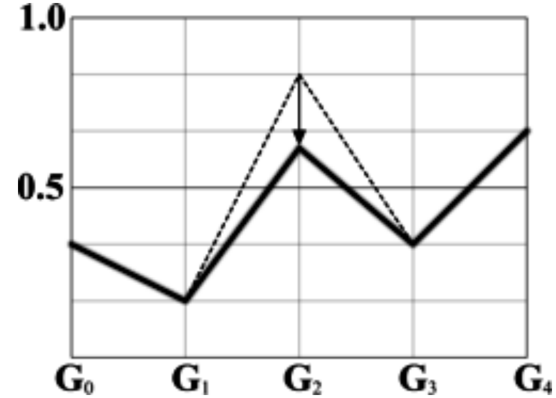
Grasshopper Galapagos - Evolutionary Principles applied to Problem Solving

Mutation Factories

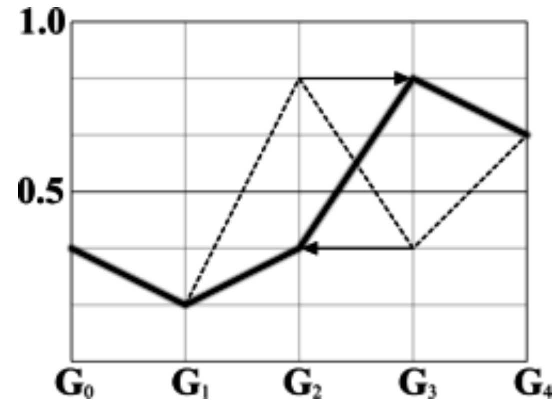
All the mechanisms we have discussed so far (Selection, Coupling and Coalescence) are designed to improve the quality of solutions on a generation by generation basis. However all of them have a tendency to reduce the diversity in a population.

The only mechanism which can introduce diversity is mutation. Several types of mutation are available in the Galapagos core, though the nature of the implementation in Grasshopper at the moment restricts the possible mutation to only Point mutations.

Point Mutation

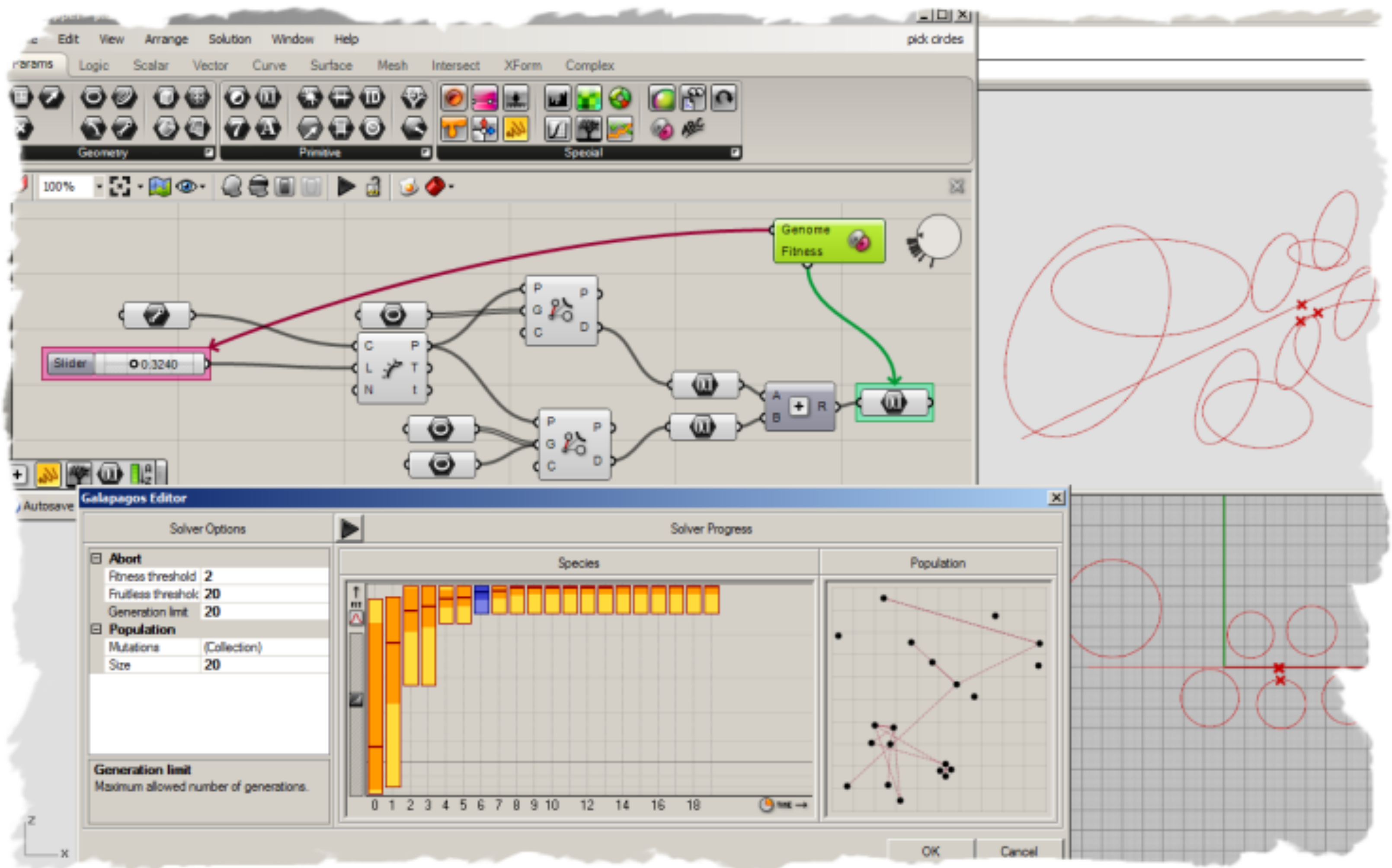


Inversion Mutation



From David Rutten's blog http://www.grasshopper3d.com/profiles/blogs/evolutionary-principles

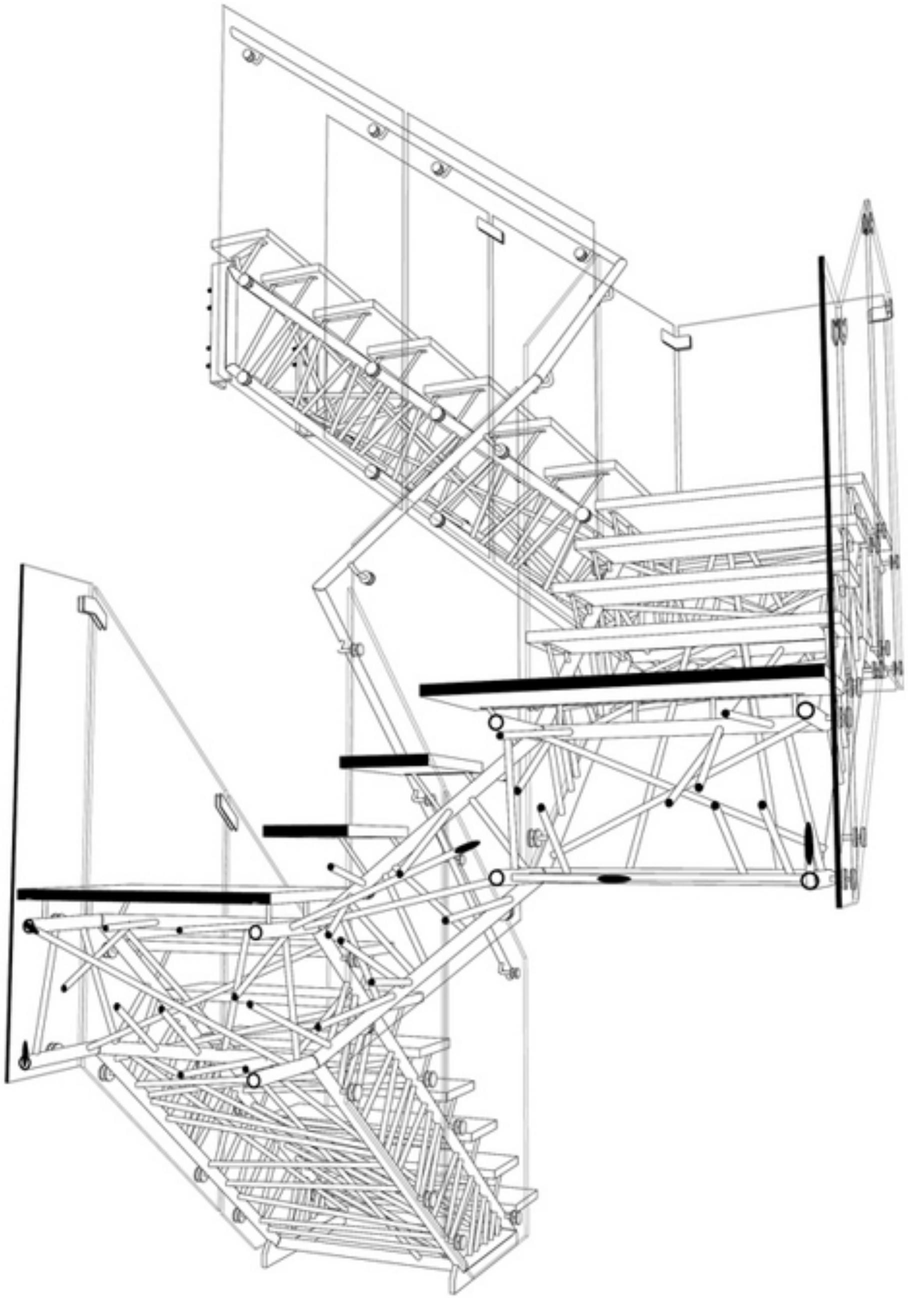
Grasshopper Galapagos - Evolutionary Principles applied to Problem Solving



From http://www.grasshopper3d.com/group/galapagos

Watch https://vimeo.com/23061345

Case Study: Genetic Stair by Caliper Studio, NYC, 2009



RHINO 4.0 + RhinoScript

makeFirstGeneration Creates one generation of completely random stair configurations.

DO until number of individuals reached

DO until allowable number of rods reached

addRandomRod
Adds random diaganol element that meets predetermined fabrication constraints and does not intersect previously created elements.

LOOP until allowable number of rods reached

exportTextFile
Converts software-specific 3D information into nodes, connections and material identities. Writes data to a newly created text file.

LOOP until number of individuals reached

TEXT FILES DESCRIBING GEOMETRIC AND MATERIAL INFO FOR EACH INDIVIDUAL IN GENERATION

CADRE Lite 3.2 + VBscript

evaluateStairConfigurations Imports text files and evaluates configurations for structural performance.

FOR EACH stair configuration in generation

importStairConfiguration
Converts text-based node and connection data to 3D geometry.

applyMaterials
Assigns material properties (strength, stiffness, etc.) to 3D members.

FOR EACH node in structural model

constrainNode
Assigns degrees of freedom, including completely fixed connections at the top and bottom.

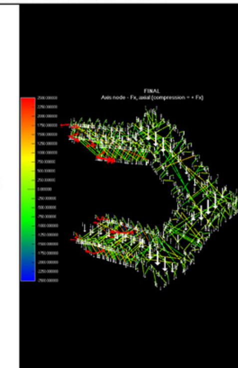
applyLoads
Assigns units of force acting on node due to dead weight and live load.

NEXT node in structural model

solveModel
Performs finite element analysis on fully constrained and loaded 3D model.

exportResults
Writes results of finite element analysis to separate text files.

NEXT stair configuration in generation



TEXT FILES DESCRIBING STRUCTURAL PERFORMANCE OF EACH INDIVIDUAL IN GENERATION

RHINO 4.0 + RhinoScript

makeNextGeneration Creates a new generation based on weighted selection of genetic material.

DO until number of individuals reached

importStair_1
Selects and imports text file of configuration from previous generation. Probability of selection is determined based on structural performance.

importStair_2
Selects and imports text file of configuration from previous generation. Probability of selection is determined based on structural performance.

assignCrossoverPoint
Location chosen at random between top and bottom of stair layout.

removeRodsFromStair_1
Rods above Crossover Point are removed from the Stair_1 configuration.

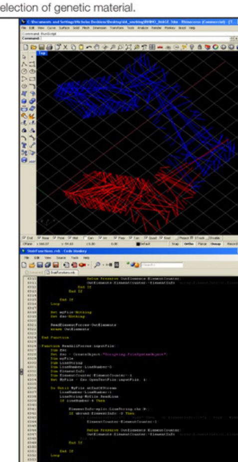
removeRodsFromStair_2
Rods below Crossover Point are removed from the Stair_2 configuration.

removeIntersectingRods
Conflicts in the new configuration that would have a negative impact on constructability are resolved with an eye towards minimal disruption to the new arrangement.

mutationFunction
Occasionally adds, removes or alters one or more rods in random fashion.

exportTextFile
Converts software-specific 3D information into nodes, connections and material identities. Writes data to a newly created text file.

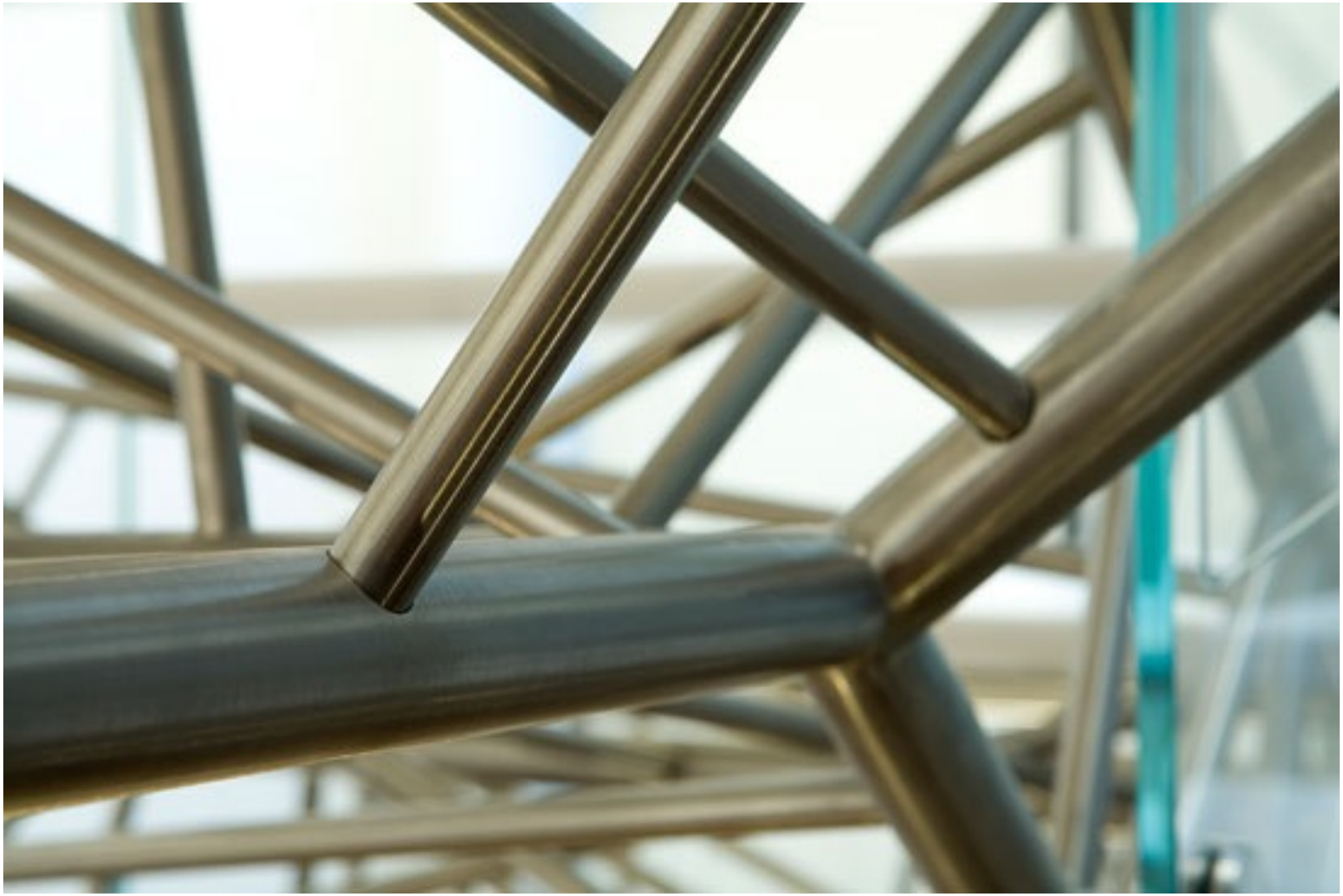
UNTIL number of individuals reached



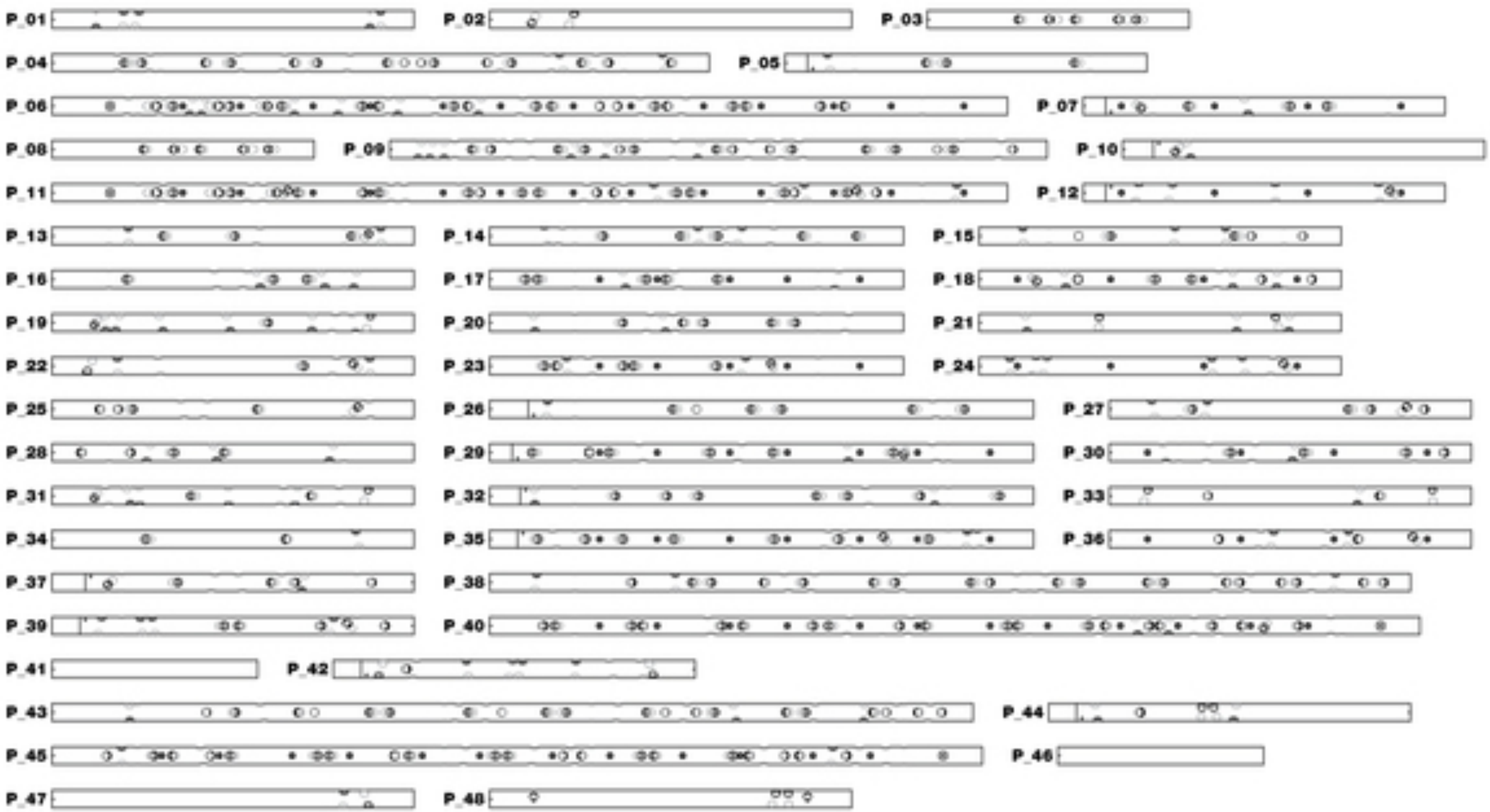
TEXT FILES DESCRIBING GEOMETRIC AND MATERIAL INFO FOR EACH INDIVIDUAL IN GENERATION

Case Study: Genetic Stair by Caliper Studio, NYC, 2009

Case Study: Genetic Stair by Caliper Studio, NYC, 2009



Case Study: Genetic Stair by Caliper Studio, NYC, 2009



Case Study: Genetic Stair by Caliper Studio, NYC, 2009

Case Study: Genetic Stair by Caliper Studio, NYC, 2009