Selected applications of computational intelligence methods to optimization in architecture

Extremely Modular Systems

Institute of Fundamental Technological Research PAN
M. Zawidzki

What is an **Extremely Modular System** (EMS)

1. Uses as few types of modules as possible (ideally **one**)
2. Allows for creation of **free-form** structures (serving given **purpose**)

What is new in **EMS**?

1. The **installation** / **construction** / **deployment** **difficulty** is moved towards the **module**:

2. EMS requires **intensive computation** for assembling its (optimal) topological and geometrical configuration.

3. The **advantage** of this “complication” is the **extreme modularization** but still capability of creating **free-form** shapes.

- **Economization** of construction / reconfiguration (by **prefabrication**)
- Intelligent **mathematical modeling**

List of EMSs:

1. ***Vault-Z*** is a concept of parametric shell system for free-form multi-branch pipe-like and vault-like constructions (patent pending)

2. ***Truss-Z*** is a modular self-supporting skeletal system for creating free-form ramps and ramp networks among any number of terminals in space

3. ***Ramp-Z*** is a modification of Truss-Z. It is a modular ramp system where each module stands individually on the ground (patent pending)

4. ***Pipe-Z*** is a parametric design system which comprised of one type of module allows for creation of complex three-dimensional, single-branch structures which can be represented by mathematical knots

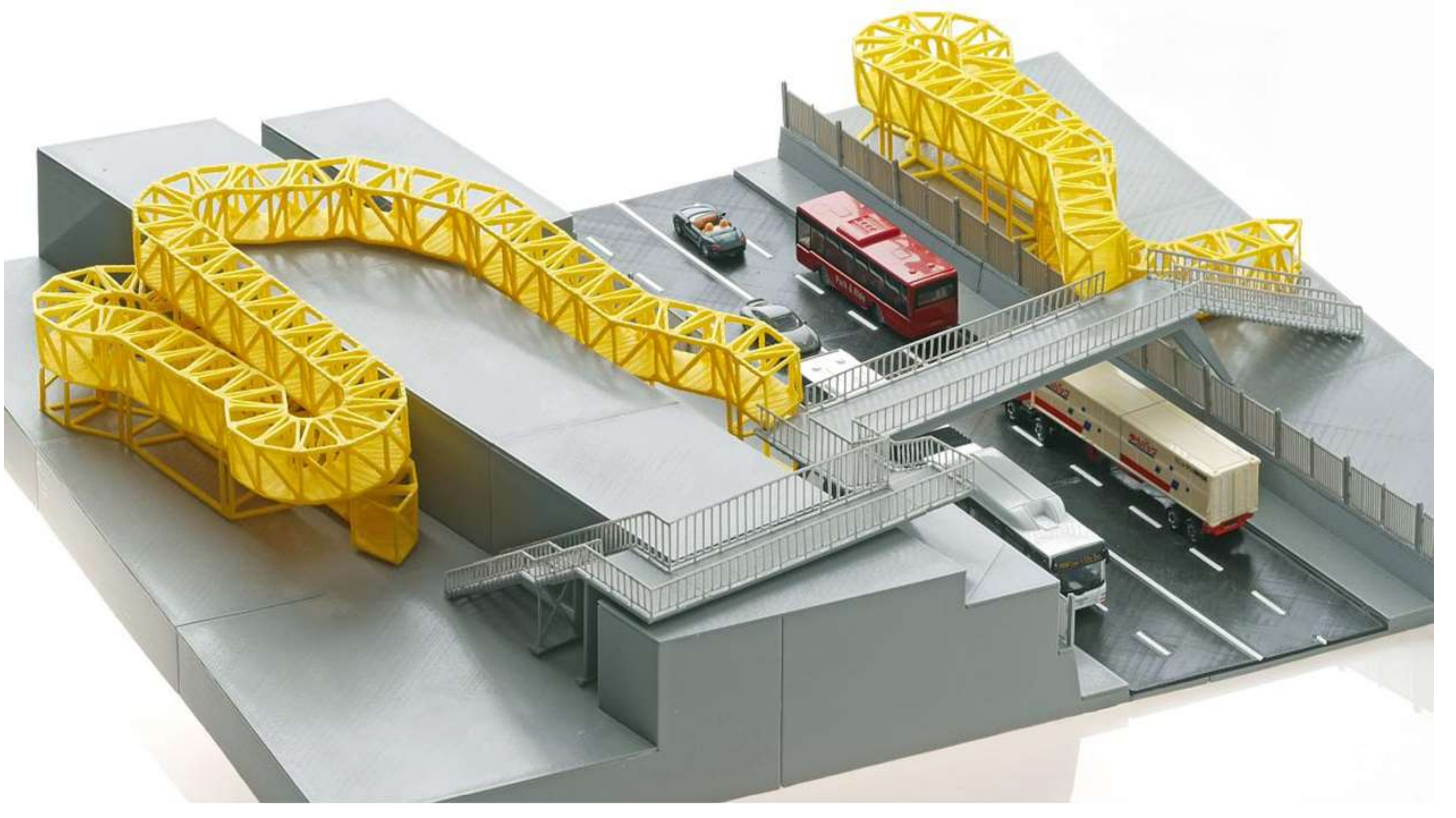
5. ***Foldable Pipe-Z*** is a modification of Pipe-Z where each module can be folded flat

6. ***Arm-Z*** is another extension of Pipe-Z. It is a concept of a hyper-redundant robotic manipulator composed of congruent modules each having one degree of freedom (1-DOF) and capable of complex movements

7. ***Multi-branch Pipe-Z*** is a recent further development of Pipe-Z allowing for construction of multi-branch pipe-like structures

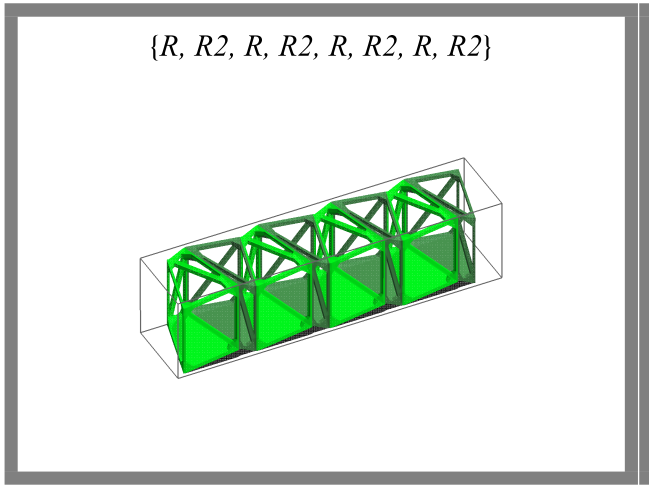
1. **Truss-Z**: Introduction

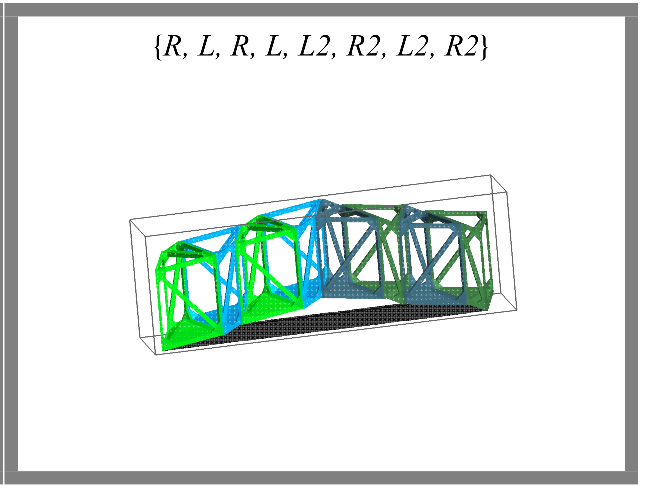
■ The principles of Truss-Z

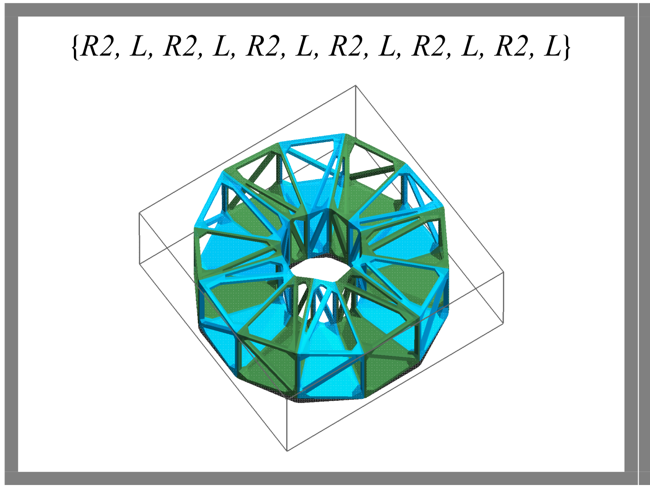


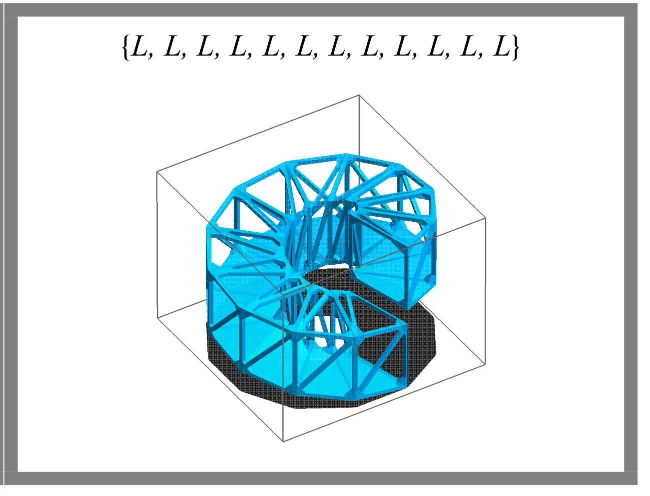
- TZ is a skeletal system for creating free-form pedestrian **ramps** and **ramp networks** among any number of terminals in space.

Basic examples of Truss-Z



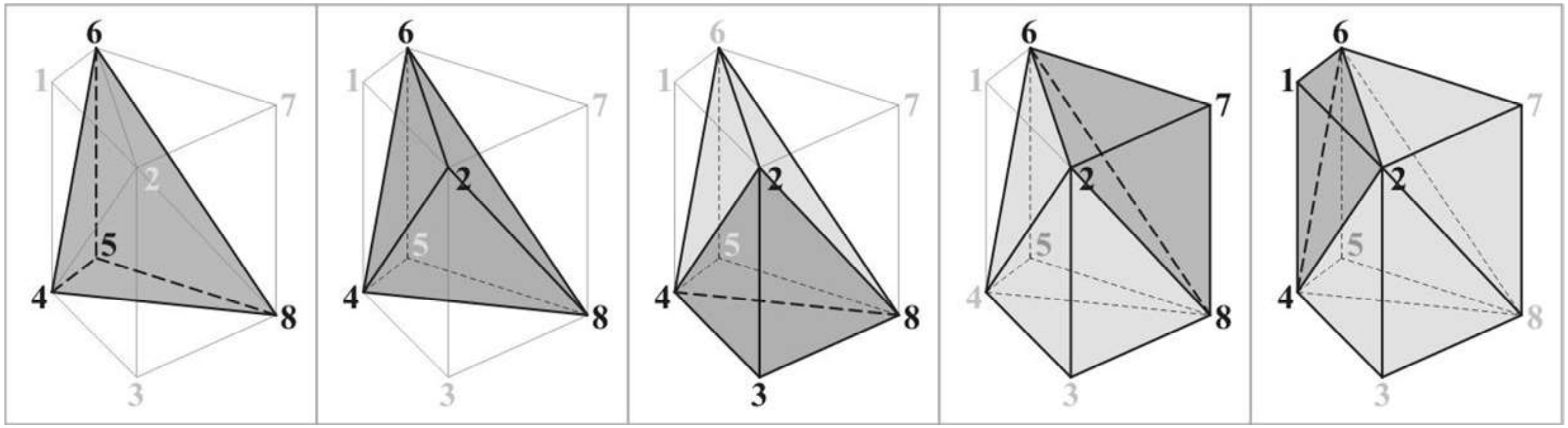






- TZ structures are composed of **four variations** of a single basic unit subjected to affine transformations (mirror reflection, rotation and combination of both)

Rigidity of TZ module according to Cauchy's Theorem

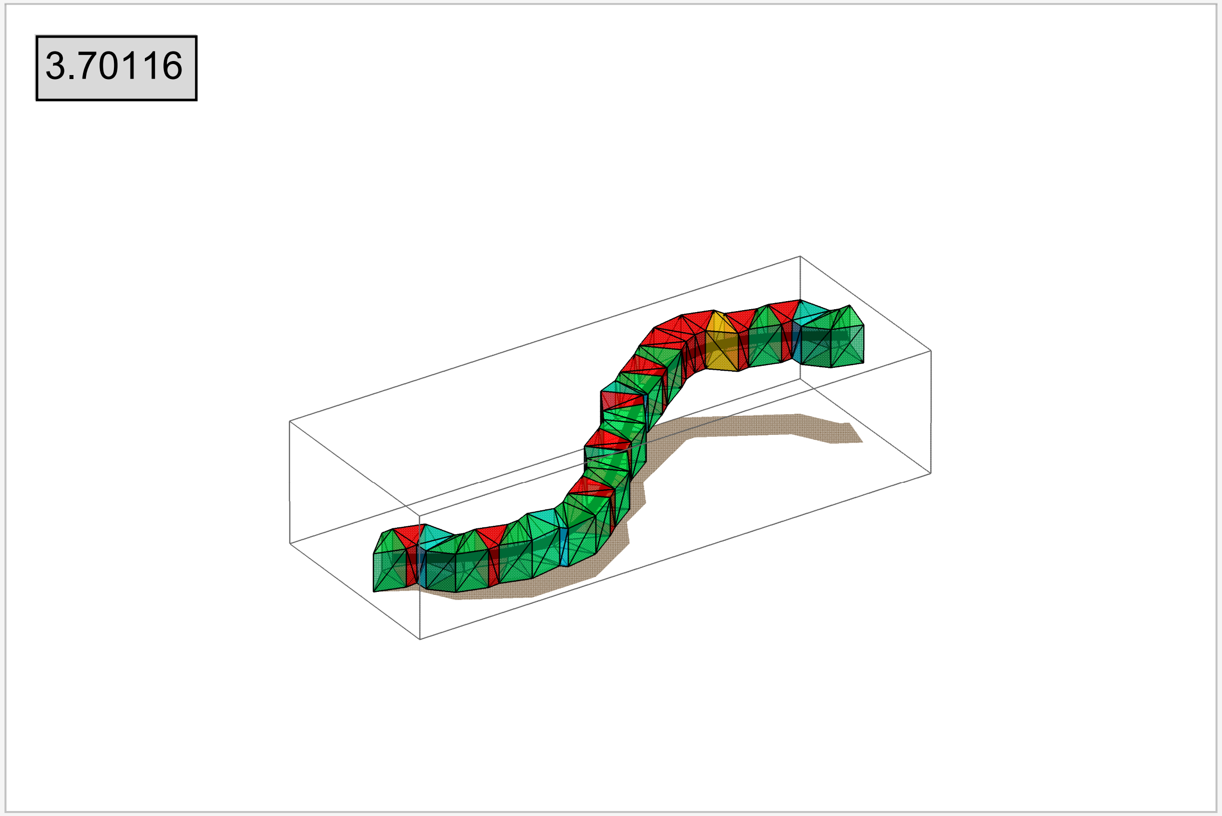


2. Assembling of **Truss-Z**: Alignment to a given path

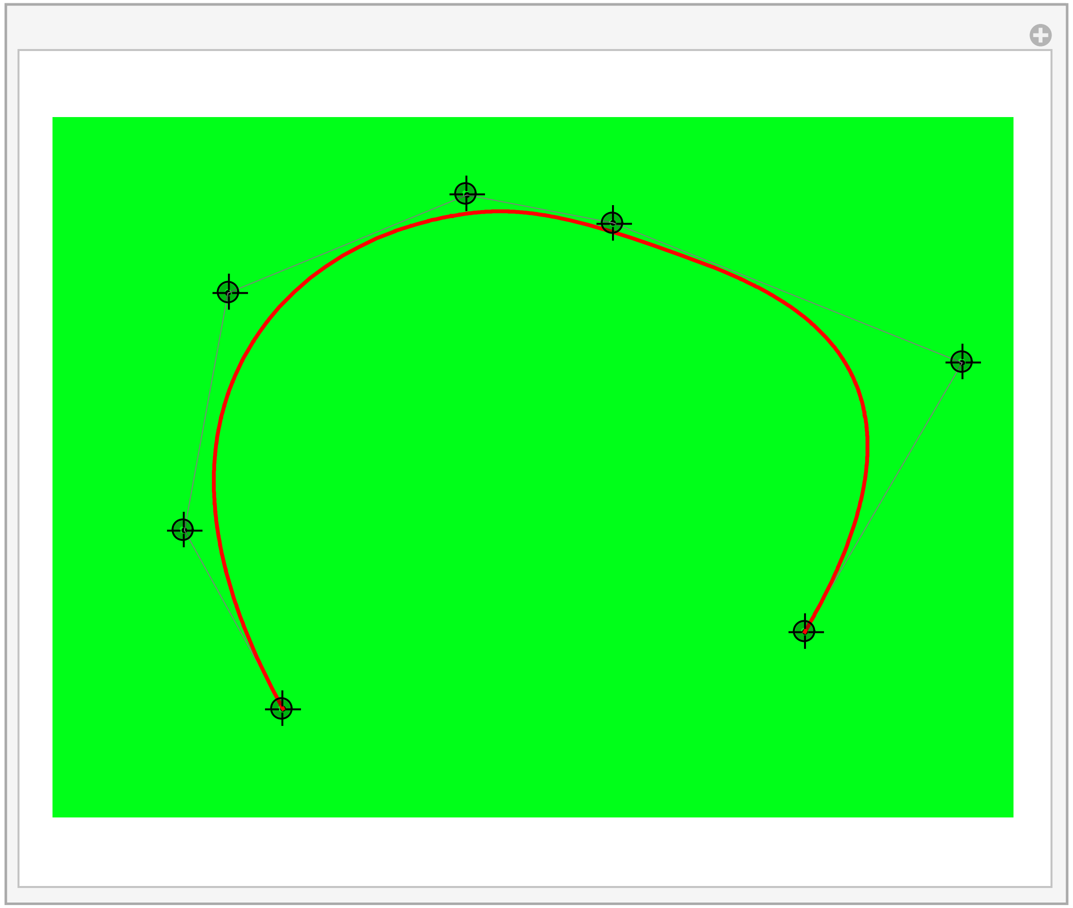
■ 3D

u: unit cetroid distance from the curve over
dot product of the unit vector and the curve

b: direction over distance

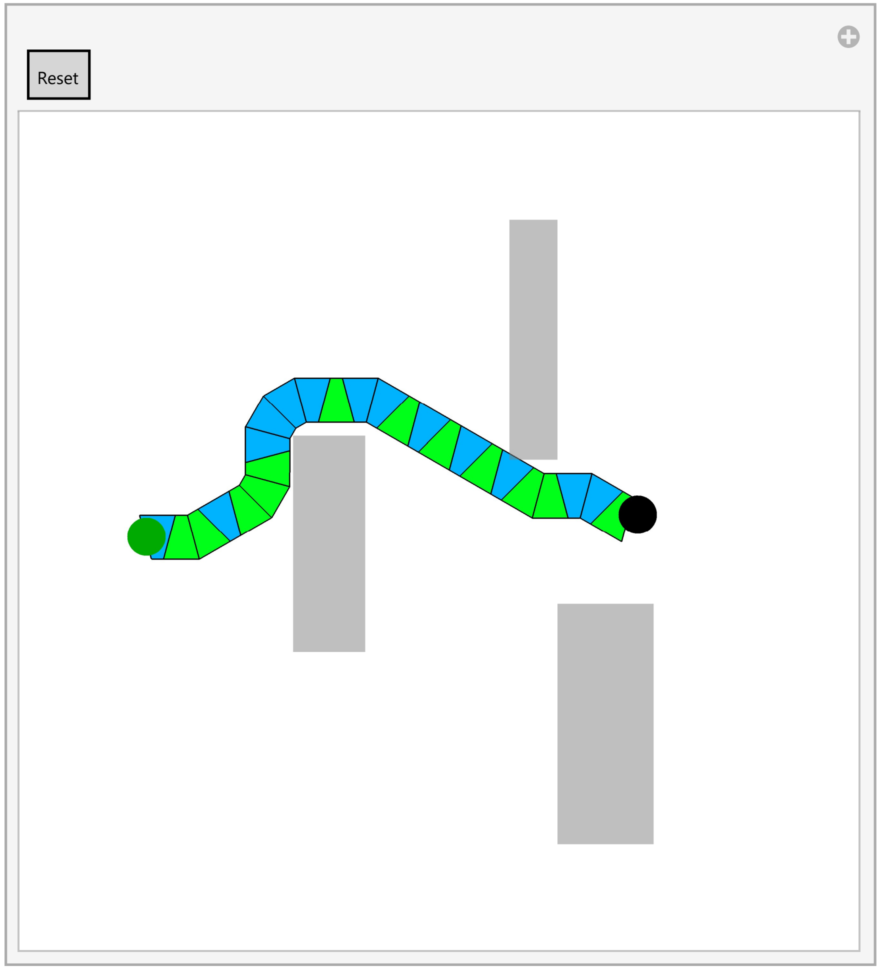


2D



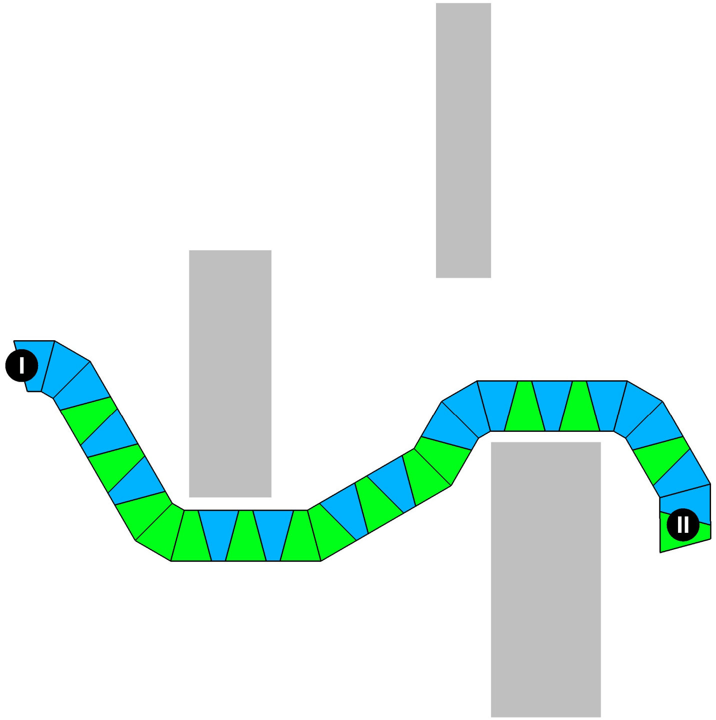
2. Assembling of **Truss-Z**: Backtracking

- Example 2D



3. **Optimization** of Truss-Z (with Evolutionary Algorithms)

- Binary representation of a TZ and the base-36 encoding



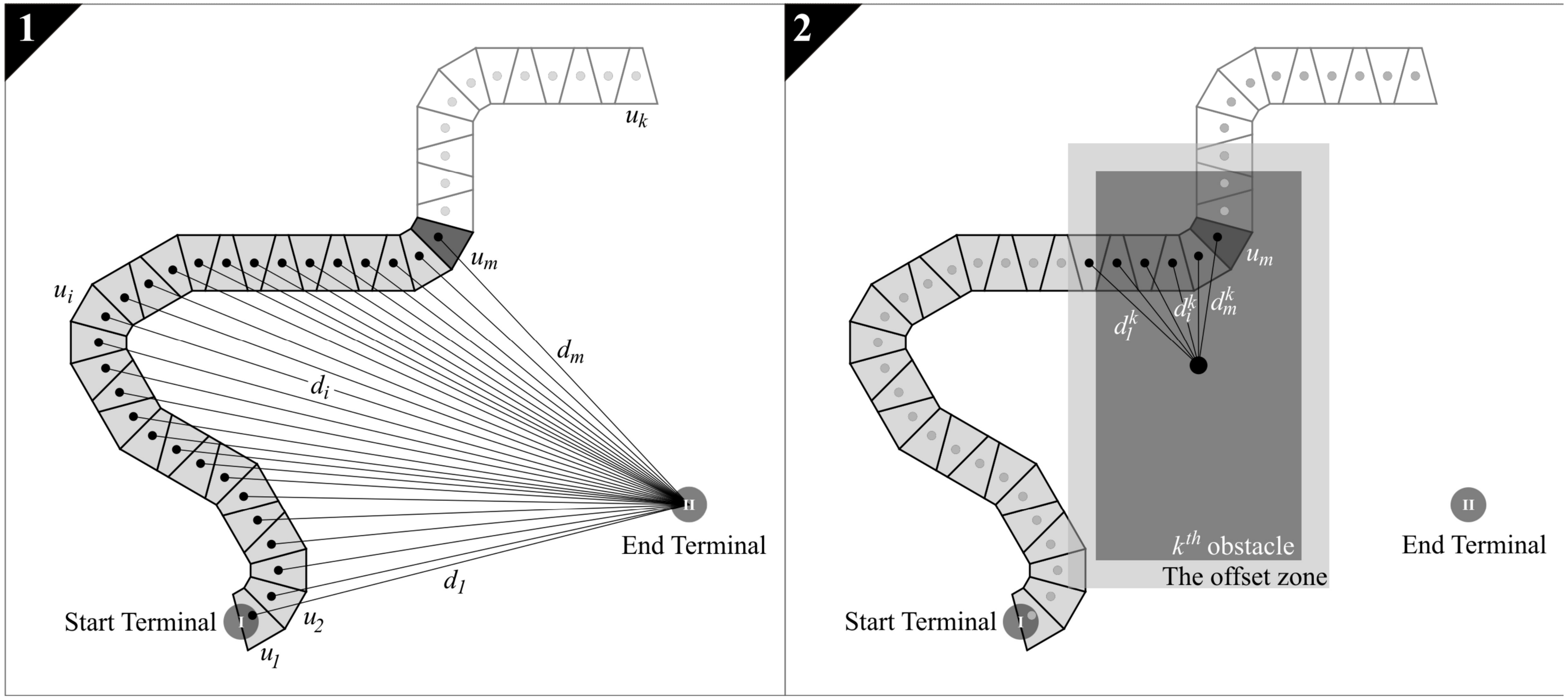
{0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1}

- Objective Function: Minimize the number of modules at minimal collisions

$$OF_s = G_s \times P_s$$

$$G_s = \frac{w_1 \sum_{i=1}^{m} d_i}{m} + w_2 \,\mathrm{Min}[d_1 \ldots d_i \ldots d_m] + w_3\, m$$

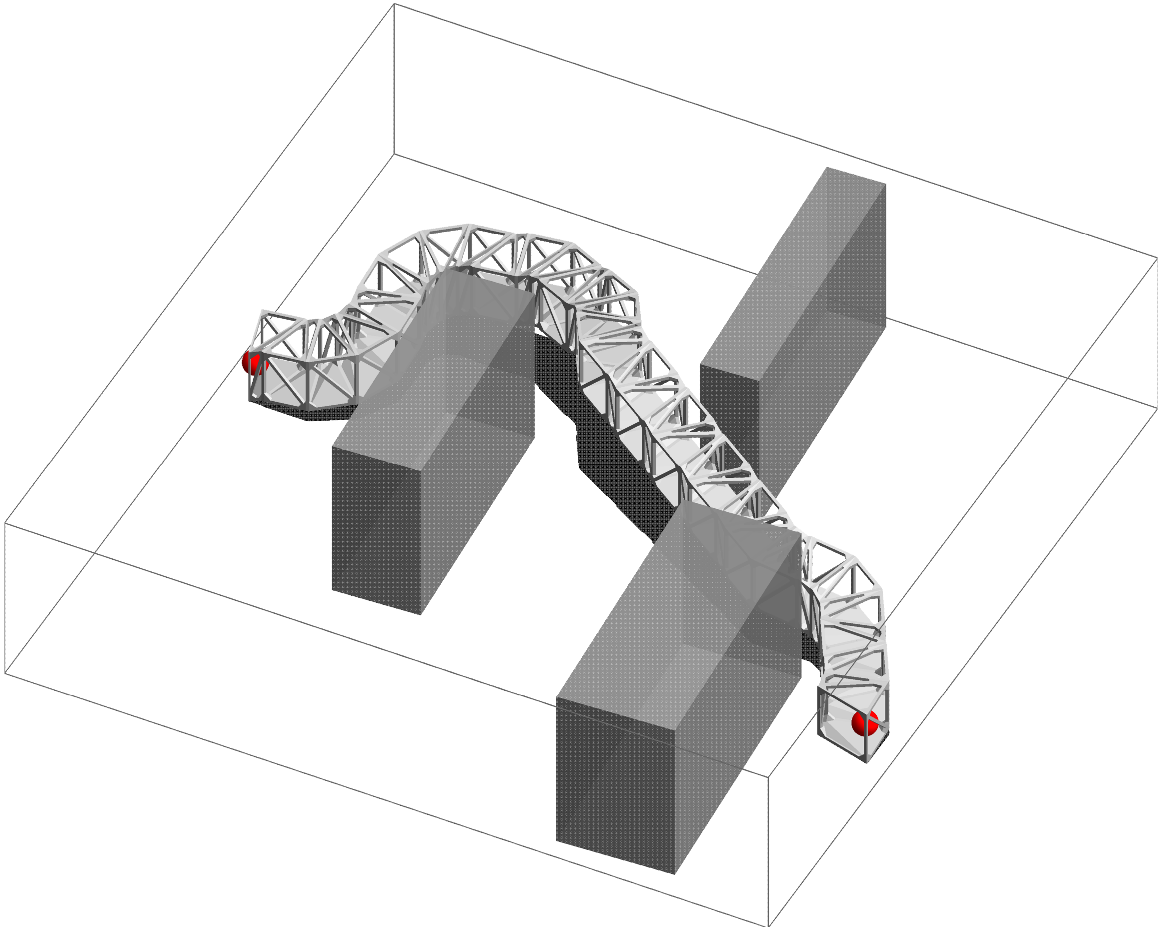
$$P_s = 1 + \frac{w_4 \sum_{k=1}^{U} c_k}{\sum_{k=1}^{U} A_k} \sum_{k=1}^{U} \frac{A_k}{\left(1 + \mathrm{Log}_{\frac{3}{2}}\left[1 + \mathrm{Min}\left[d_1^k \ldots d_i^k \ldots d_m^k\right]\right]\right)}$$



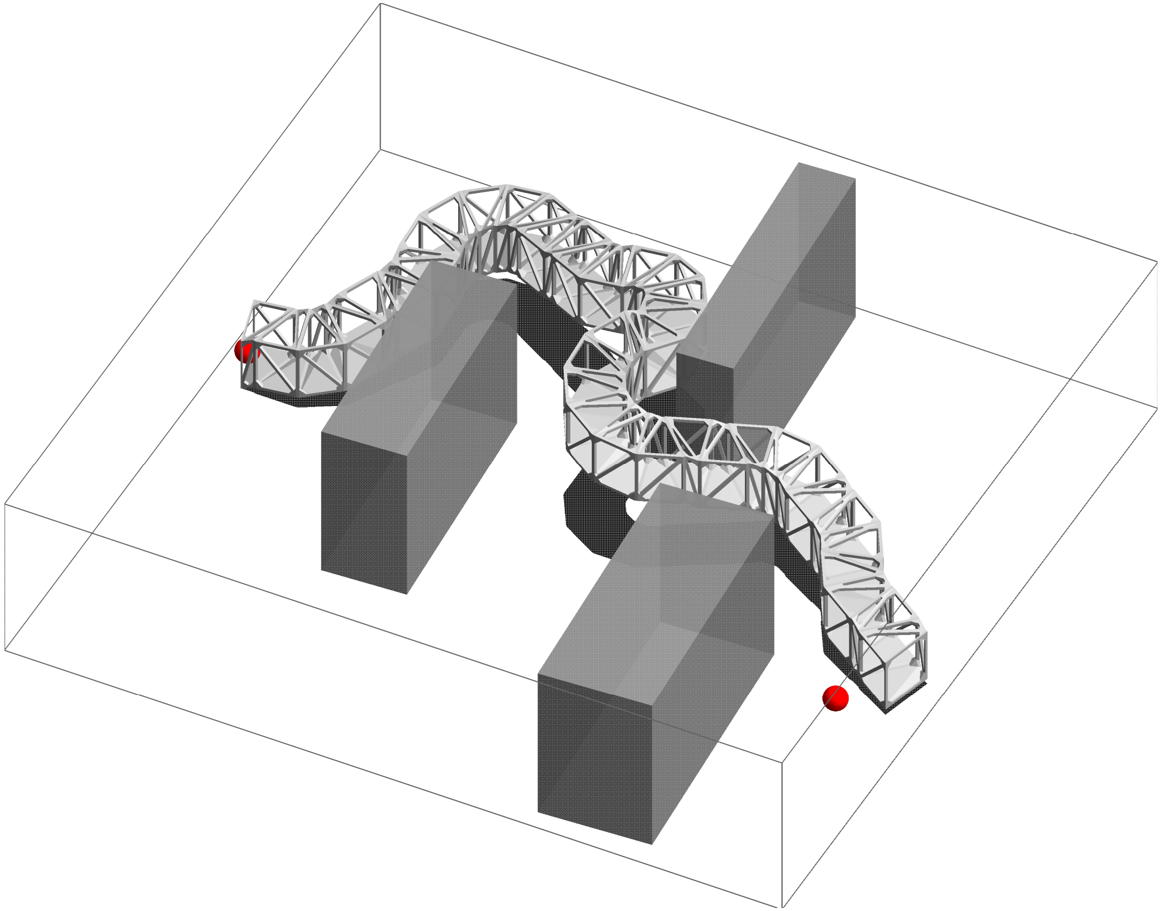
- Formulation of genetic operators: mutation and recombination

3. Implementation of **Evolution Strategy** and **Genetic Algorithm**

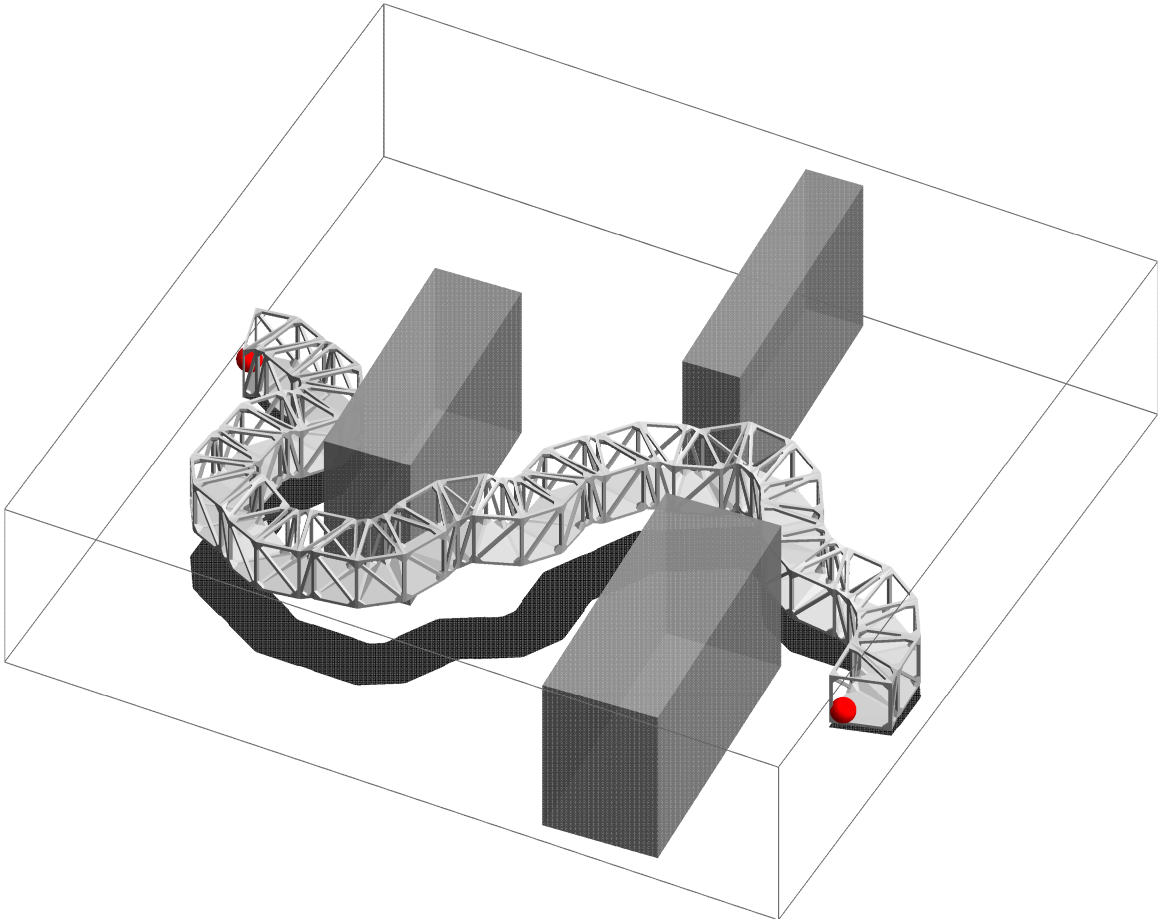
- "Manual" solution



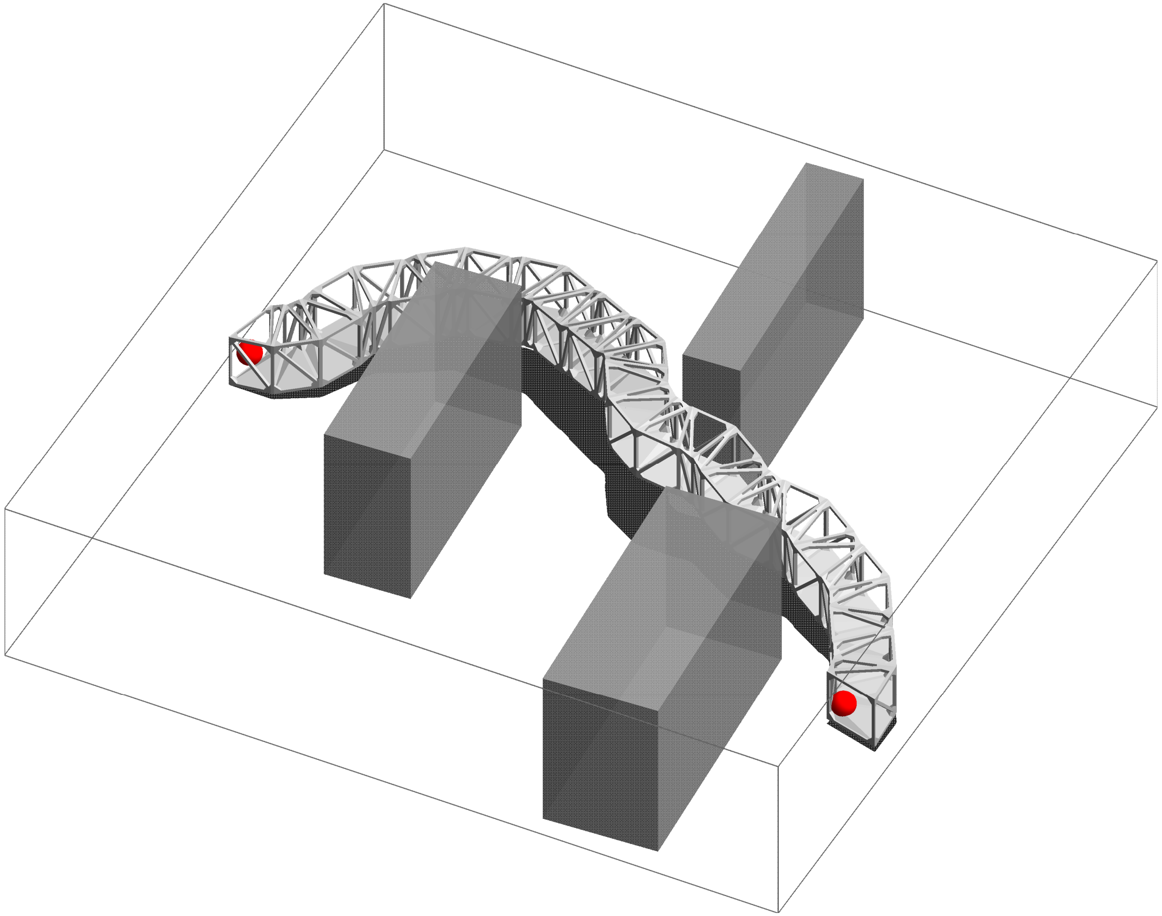
The best Random Search



The best Evolution Strategy

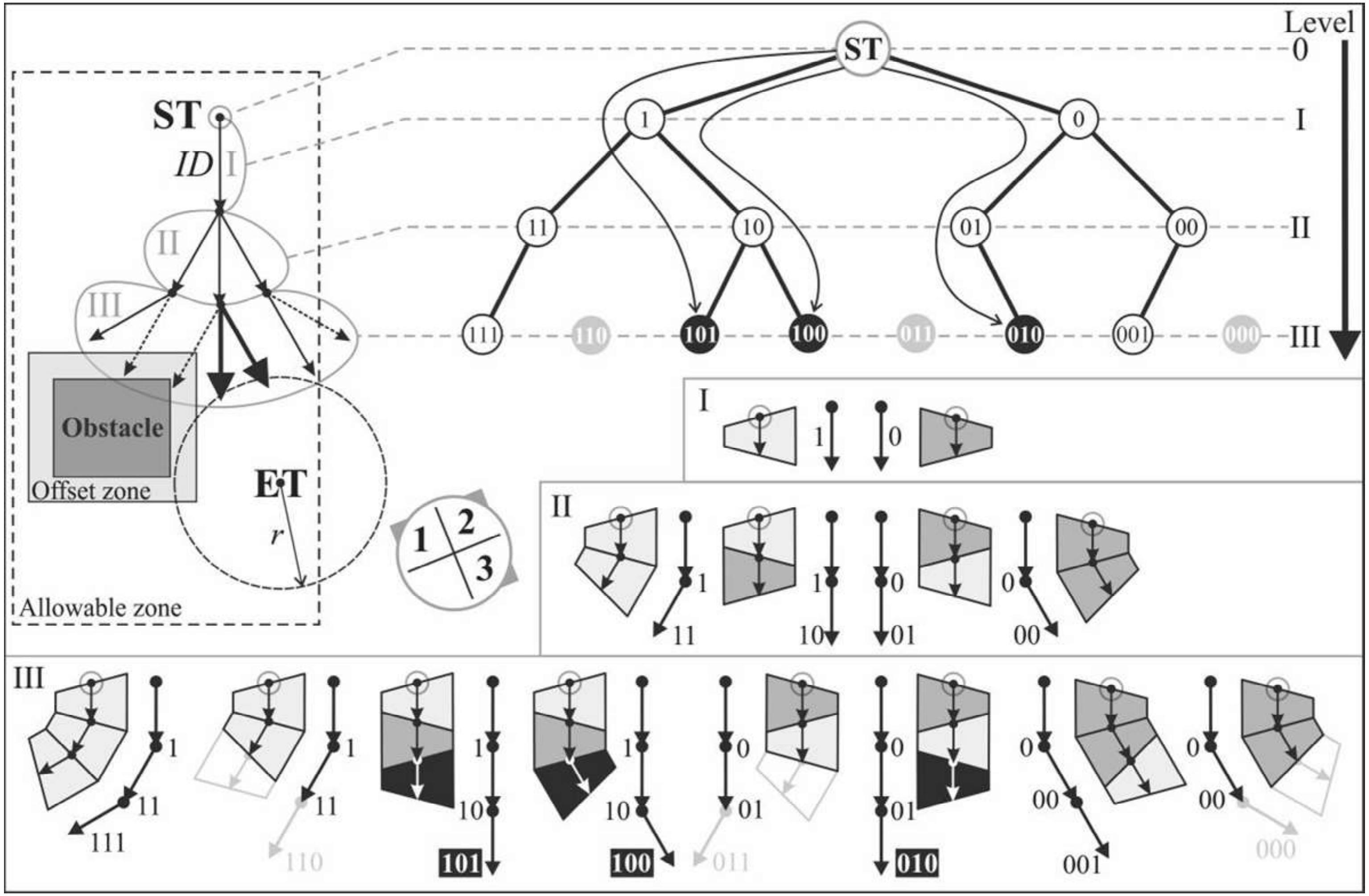


- The best OPX (better than the “manual”)

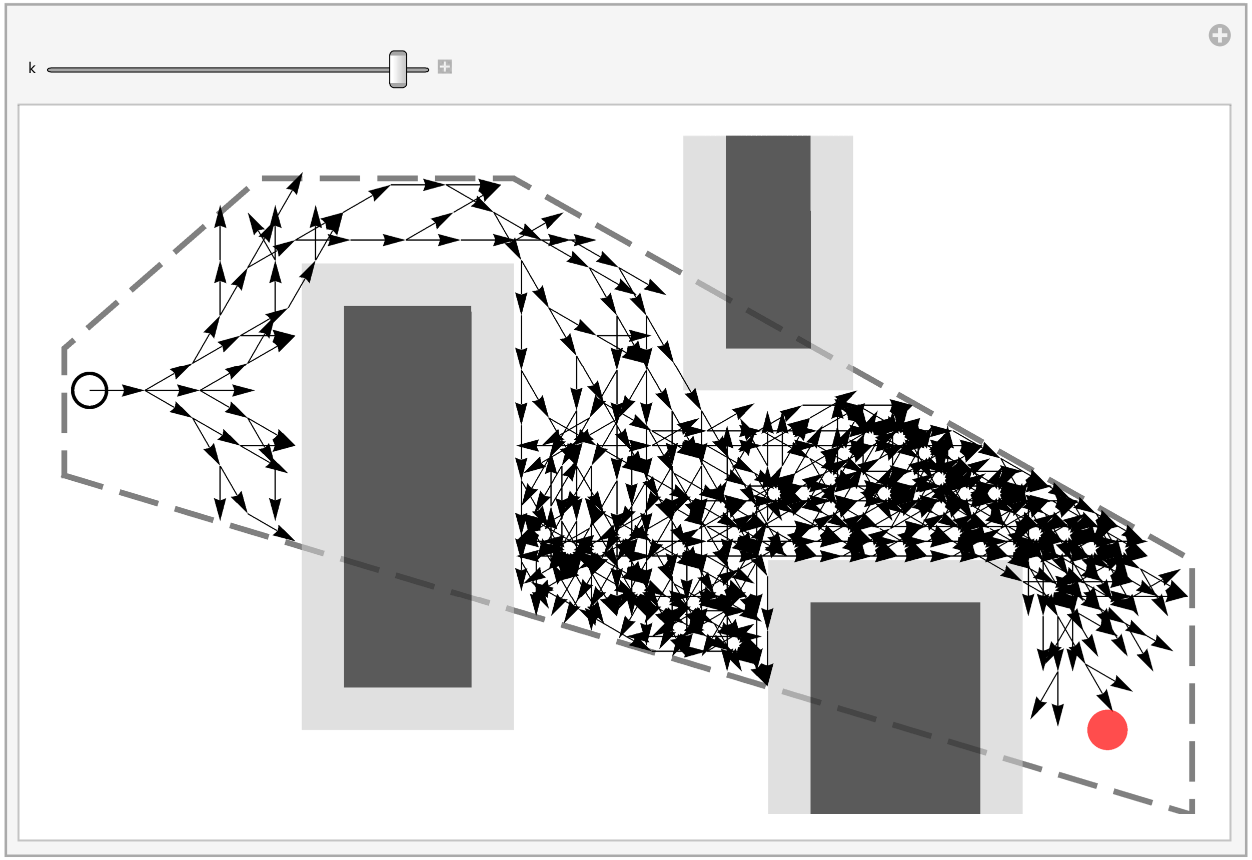


4. Optimization of Truss-Z with **graph-theoretic** approach

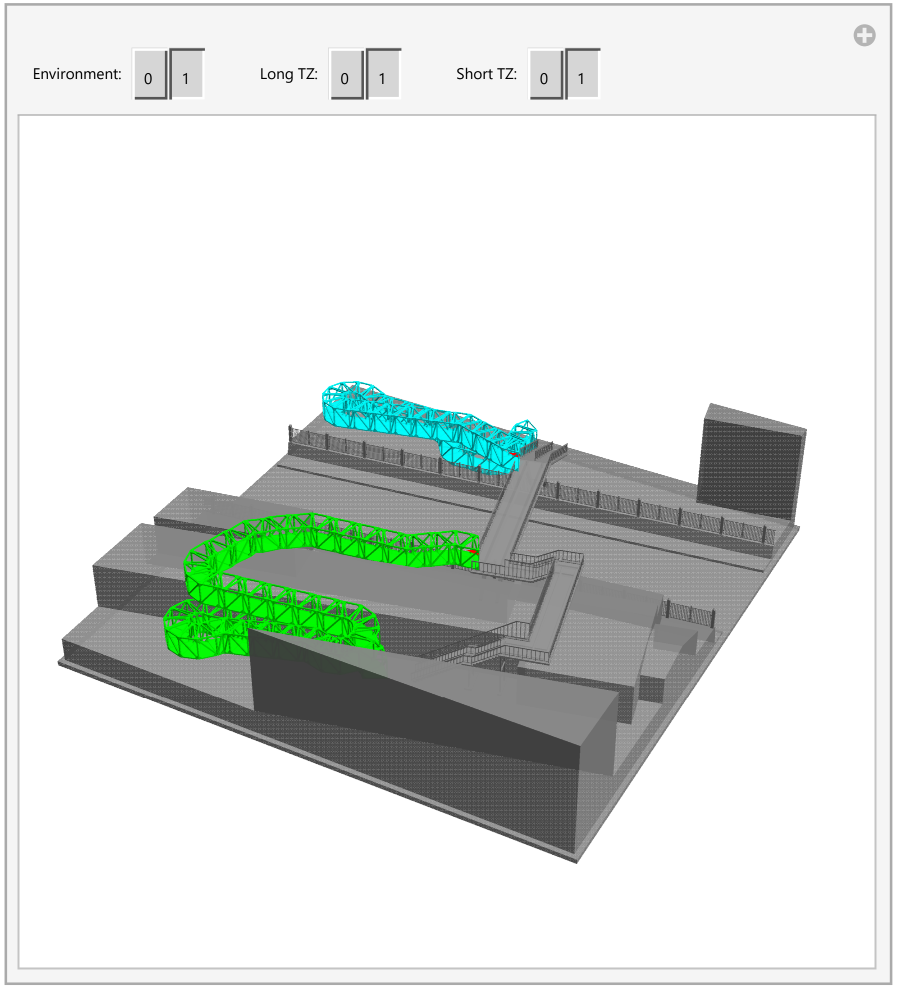
- TZ as a path-graph (degenerate tree)



Space exploration by the Search Space Tree (SST) - a binary tree

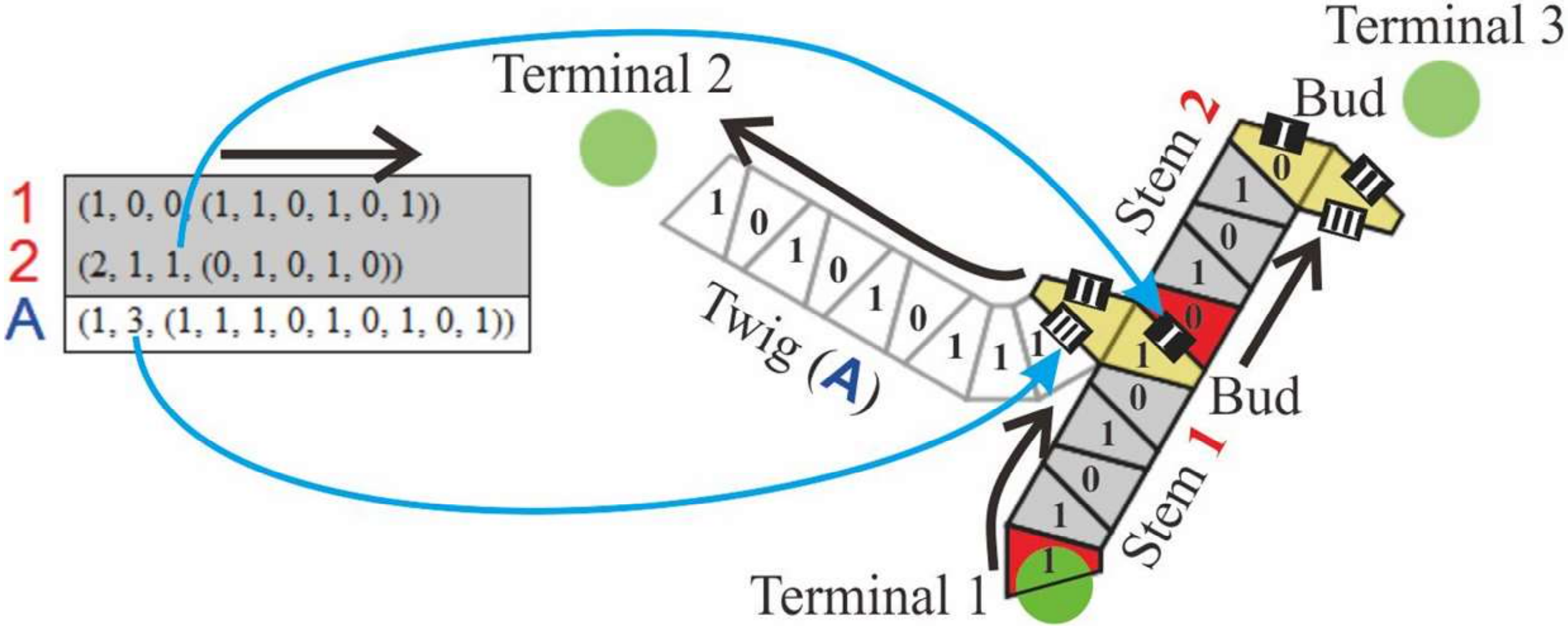


- Introduction of additional criteria: “Geometrical Simplicity”, “Number of Turns”
- Case-study: retrofitting of an existing overpass



5. Optimization of **Multi-branch** Truss-Z (with evolution strategy)

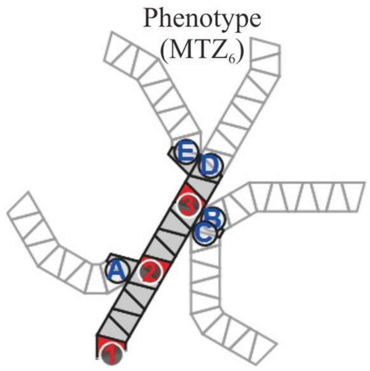
- The principles of multi-branch Truss-Z (MTZ)



- New operators: adding/removing/displacing branches (stems and twigs), adding/removing/inverting units at branches

Start	Transformation	Operator
General structure	⮧	
↓	Add a *stem:* Add *twigs:* Remove *branches:*	$aS\ [G_h, (s_i, BF_i, u_i)]$ $aT\ [G_i, ((p_1, BF_1, u_1),(p_2, BF_2, u_2),\ldots,(p_k, BF_k, u_k))]$ $rB\ [G_i, ((p_1, BF_1), (p_2, BF_2), \ldots,(p_i, BF_k))]$
Substructure @ buds	⮠ ⮧	
↓	Displace *branches*:	$dB[G_i, (((p_i, BF_i), (p_j, BF_j)),\ldots)]$
Units @ branches	⮠ ⮧	
↓	Add units @ *branches*: Remove units @ *branches*: Invert units @ *branches*:	$aU[G_i, ((p_1, BF_1, (v_{1^1}, l_{1^1}),\ldots, (v_{k^1}, l_{k^1})),\ldots, (p_j, BF_j, (v_{1^j}, l_{1^j}),\ldots, (v_{k^j}, l_{k^j})))]$ $rU[G_i, ((p_1, BF_1, (l_{1^1},\ldots, l_{k^1})),\ldots, (p_j, BF_j, (l_{1^j},\ldots, l_{k^j})))]$ $iU[G_i, ((p_1, BF_1, (l_{1^1},\ldots, l_{k^1})),\ldots, (p_j, BF_j, (l_{1^j},\ldots, l_{k^j})))]$
End	⮠	

Phenotype ← Genotype → Tabulated Genotype



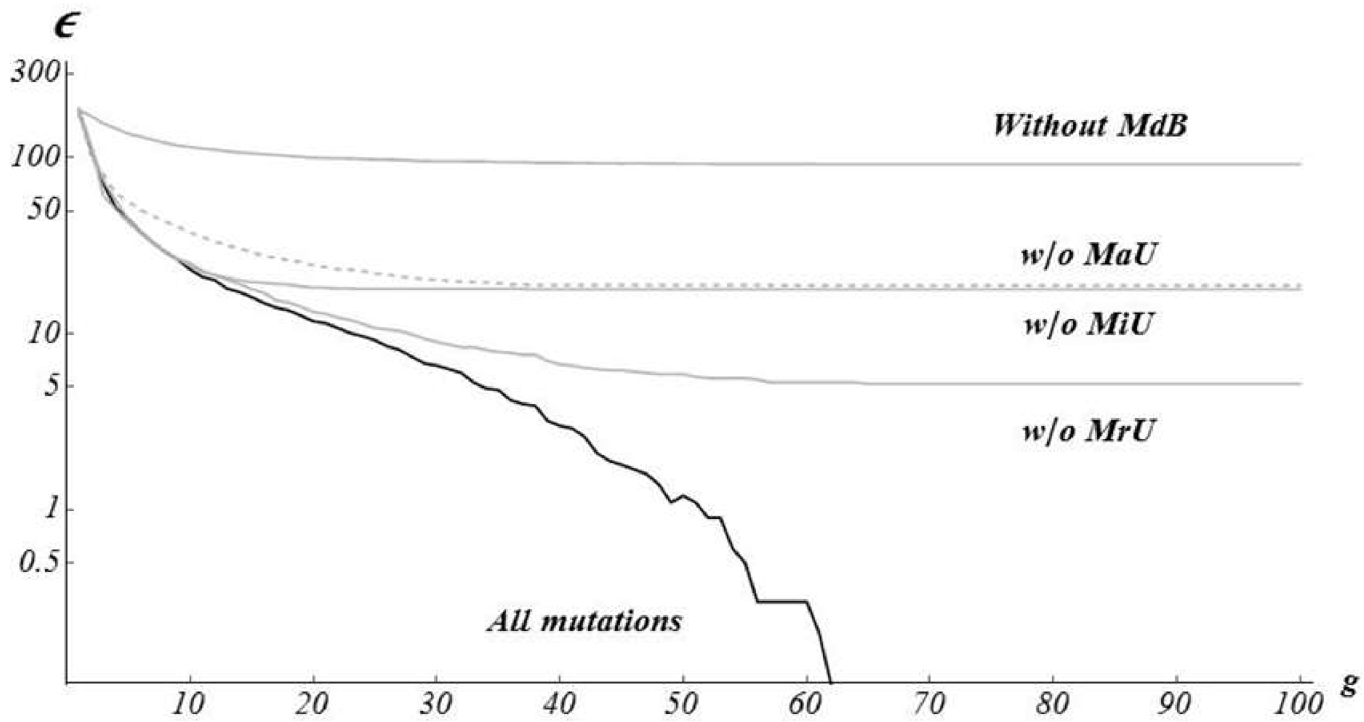
Genotype (G_6)

```
{1, 0, 0, {1, 1, 0, 1, 0, 1}}
{2, 1, 1, {0, 1, 0, 1, 0}}
{3, 2, 1, {1, 0, 1}}
{1, 3, {1, 1, 1, 0, 1, 1, 0, 1, 0}}
{2, 2, {1, 0, 1, 1, 0, 1, 0, 1, 0, 1}}
{2, 3, {0, 1, 0, 1, 0, 0, 0, 1, 0, 1}}
{3, 1, {0, 1, 0, 1, 0, 1, 0, 0, 1}}
{3, 2, {0, 0, 1, 0, 1, 0, 0, 1, 0}}
```

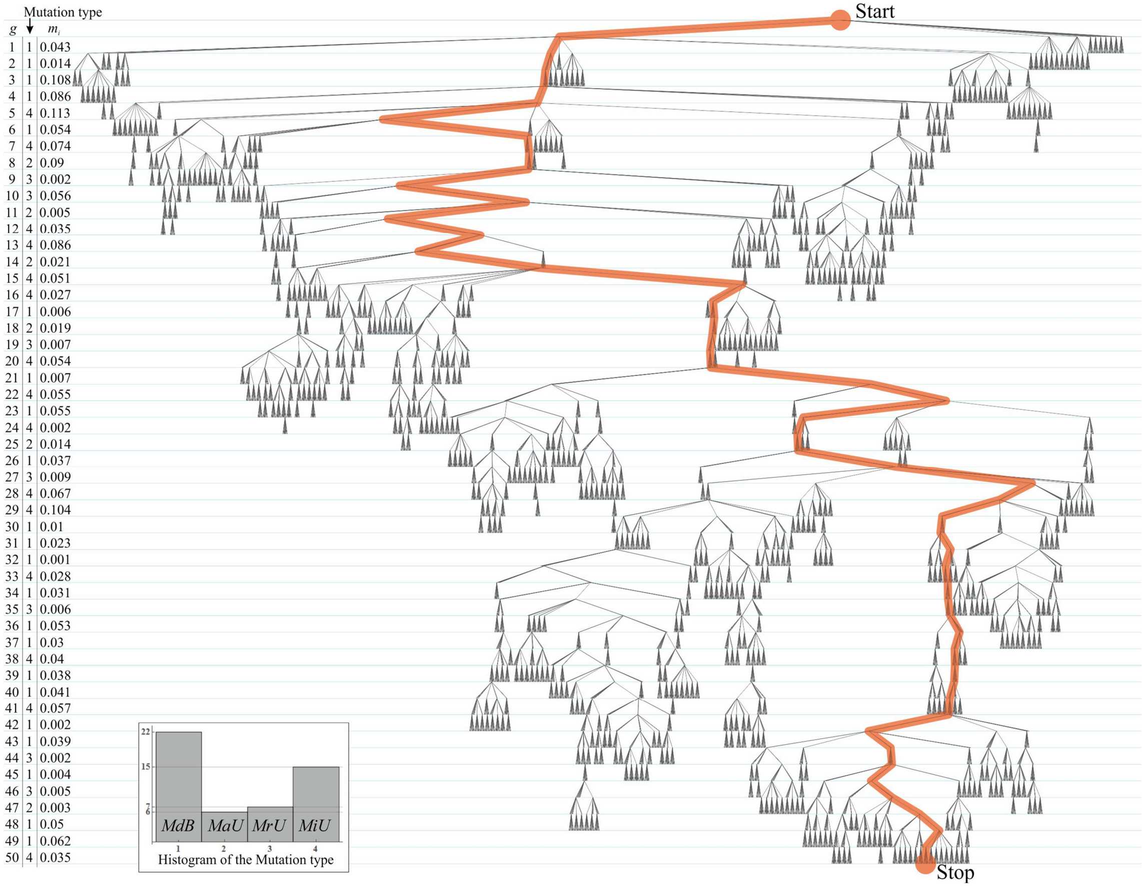
Tabulated genotype ($T[G_6]$)

	Bud ID ↓ Bud face:	I	II	III
Stems	(0)	{1, 1, 0, 1, 0, 1}	x	x
	1	{0, 1, 0, 1, 0}	x	x
	2	{1, 0, 1}	x	x
Twigs	1	x	x	{1, 1, 1, 0, 1, 1, 0, 1, 0} (A)
	2	x	{1, 0, 1, 1, 0, 1, 0, 1, 0, 1} (B)	{0, 1, 0, 1, 0, 0, 0, 1, 0, 1} (C)
	3	{0, 1, 0, 1, 0, 1, 0, 0, 1} (D)	{0, 0, 1, 0, 1, 0, 0, 1, 0} (E)	x

- All 4 types of mutations necessary: displace-branch MdB, add-unit MaU, remove-unit MrU, invert-unit MiU

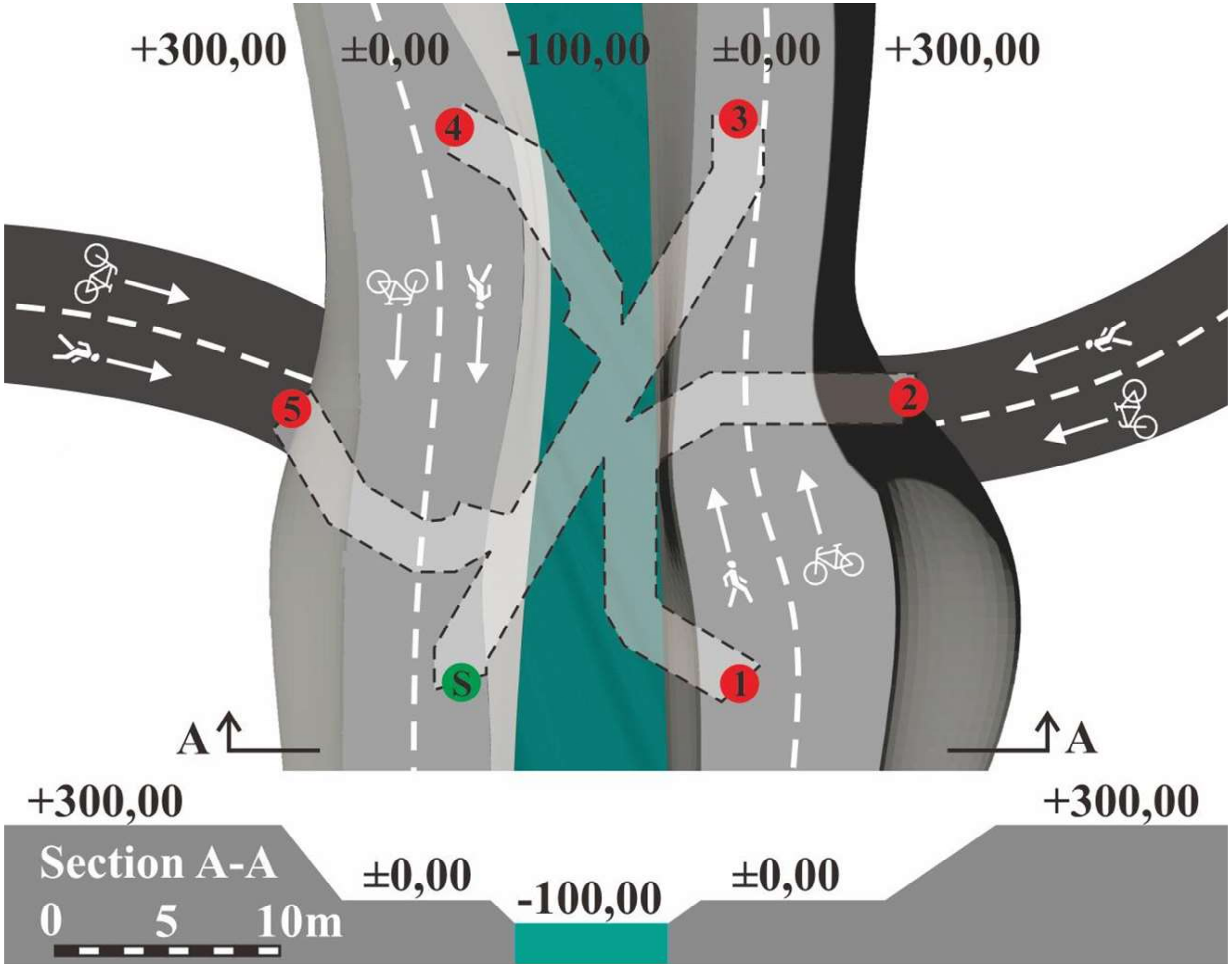


- Quasi-optimization (the ideal solution is known). Only successful offspring is shown.



5. **Multi-branch** Truss-Z: case-study

- The site for the 6-branch TZ (MTZ_6)

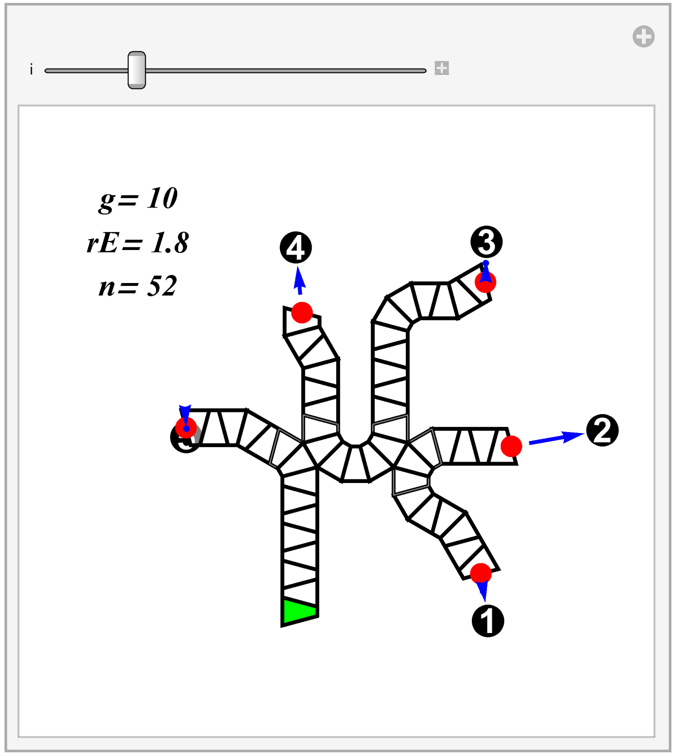


- Objective Function: “reaching error” *rE* for each twig and number of units to be minimal

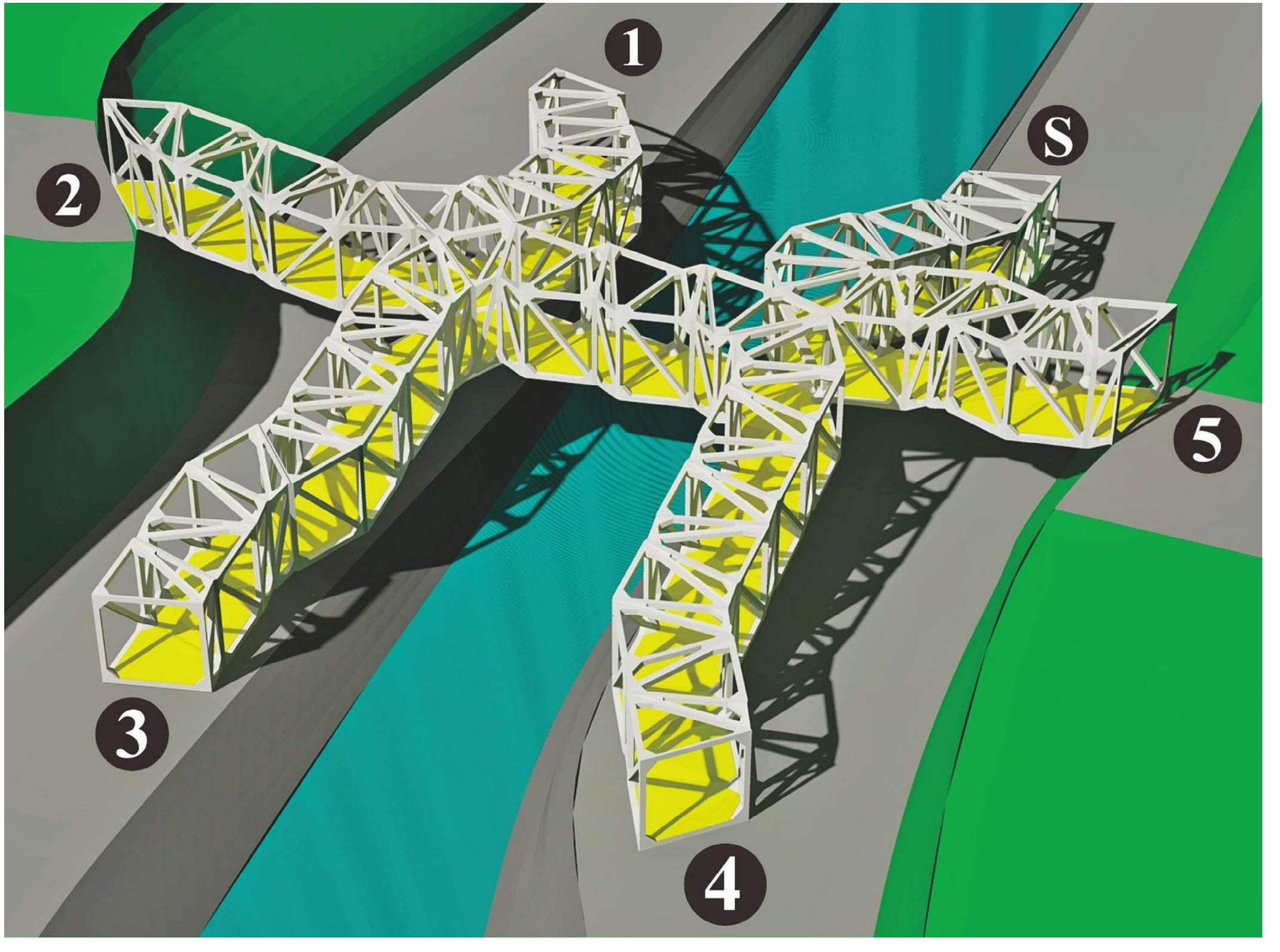
$$OF_{MTZ} = n + w \times r_E$$

$$r_E = \sum_{i=1}^{5} \epsilon_i$$

- Selected generations of the 2nd ES trial with $w = 4$.

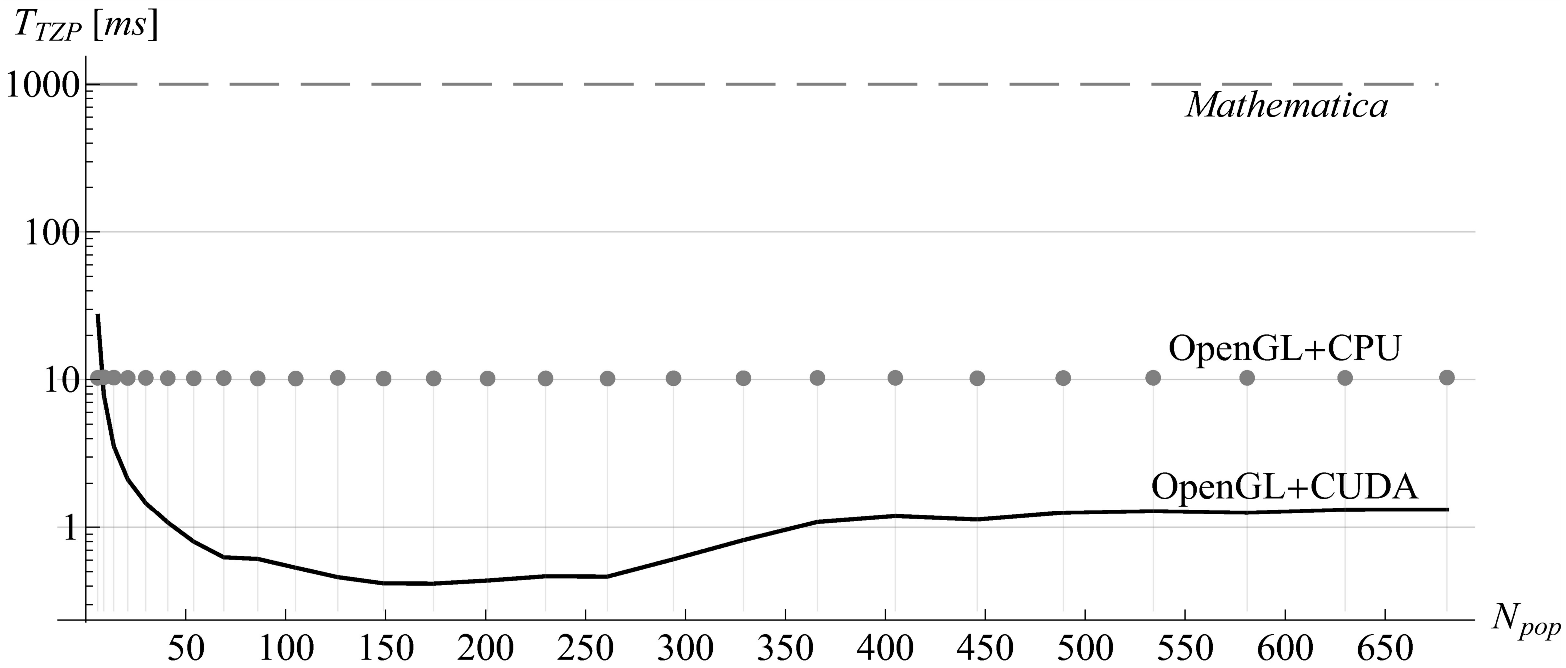


Visualization of the best result



6. Effective multi-objective optimization of Truss-Z with GPU

- Introduction of TZ self-intersection prohibition
- Introduction of formal cost matrix *C* which allows for optimization of a TZ in environment ***E*** where obstacles can have any shape and cost expressed by a real number. Full advantage of well-established **image processing** tools can be used.
- The introduction of cost flow matrix *F* substantially improves the efficiency of the algorithm. Good solutions can be found consistently and quickly also in highly constrained environments.

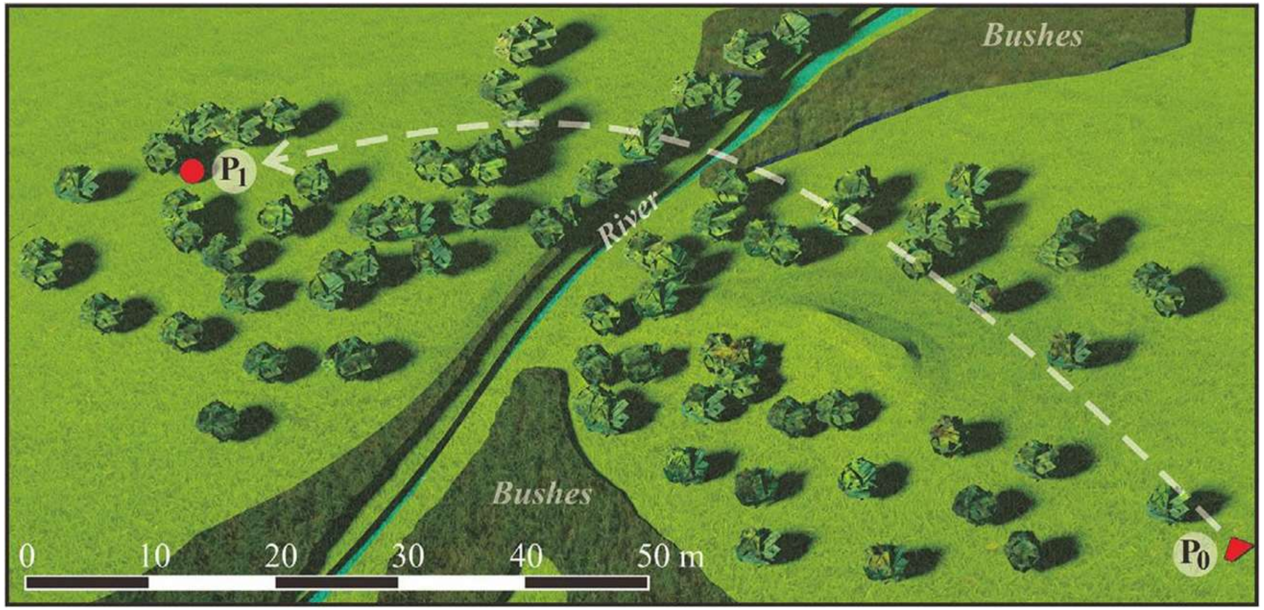

T_{TZP} [ms]
1000
100
10
1
Mathematica
OpenGL+CPU
OpenGL+CUDA
50
100
150
200
250
300
350
400
450
500
550
600
650
N_{pop}

- A typical genetic algorithm (GA) is applied, with three operators : mutation, crossover, and tournament selection.
- The length of entire TZ is unknown beforehand and it becomes a new variable also encoded in a genotype.
- Efficient implementation utilizing GPU is introduced, where hundreds of individuals are drawn onto a single image in card's memory and evaluated in parallel.
- This framework can handle <u>any number of TZs</u> in one environment.

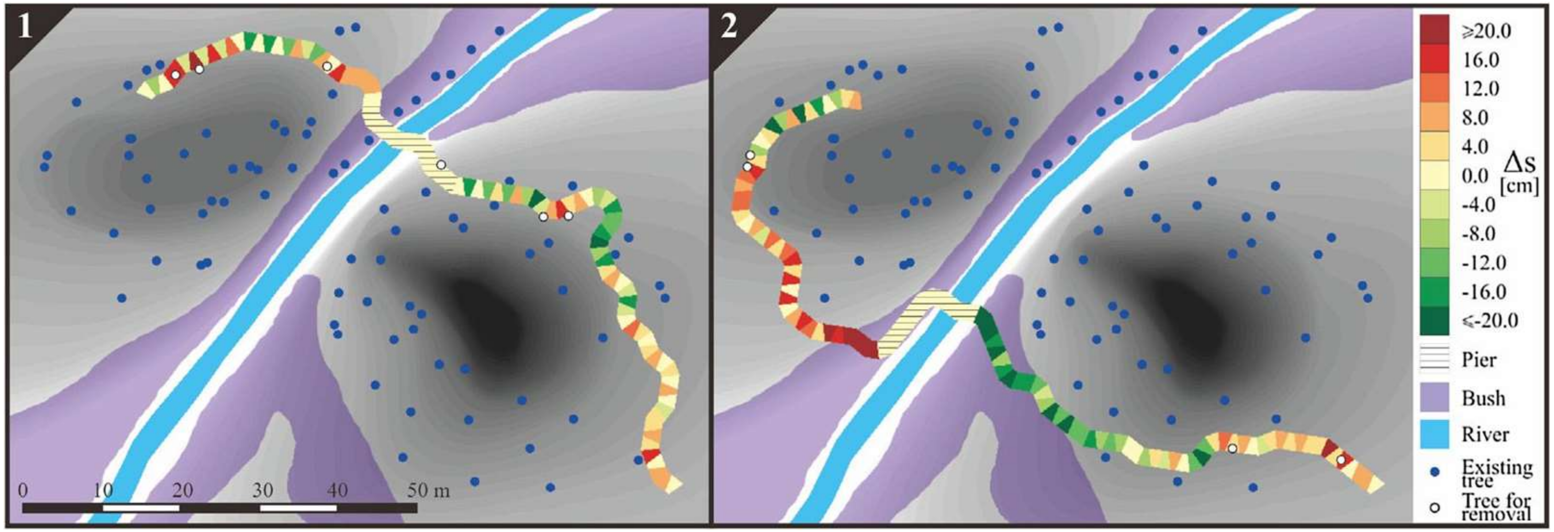
6. Optimization of Truss-Z with GPU. **Case-study 1**.

"Modular path" objective

- Paving of trapezoidal path from P_0 to P_1 in a hilly environment with a river. Two types of vegetation (trees and bushes).
- Max. elevation difference between two units is 20 cm.
- Minimize earthwork and other costs.

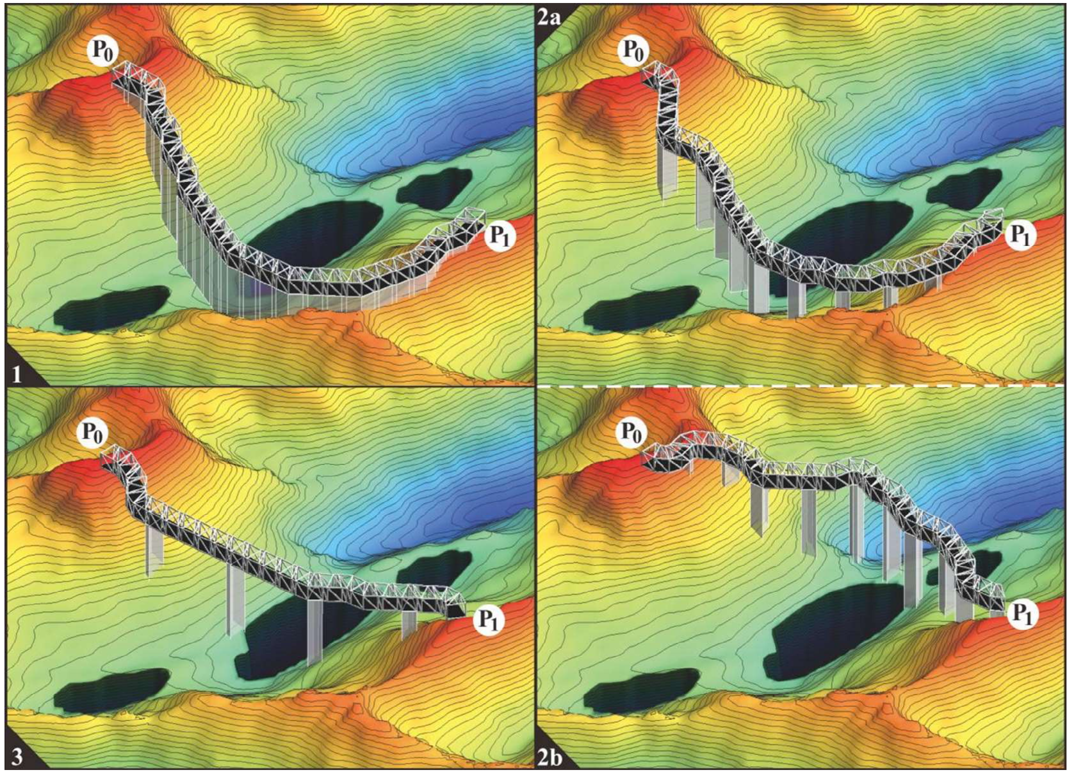


- The best and second-best results (1: 92 units, 6 trees removed; 2: 106 units, 4 trees removed).



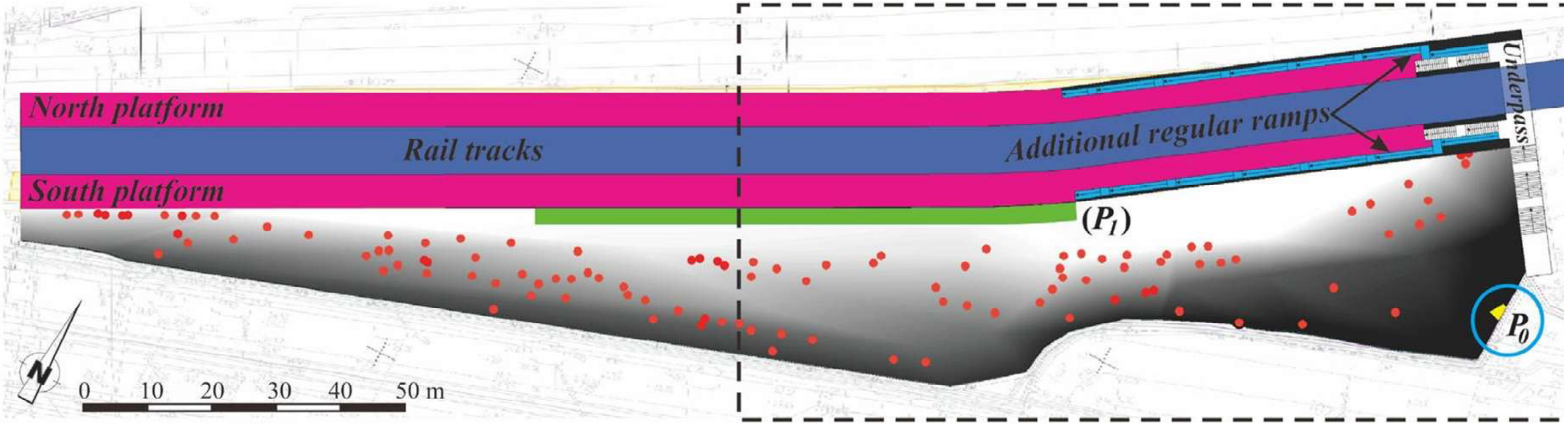
6. Optimization of Truss-Z with GPU. **Case-study 2.**

- "Mountain pier"
  - Create TZ from P_0 to P_1 with fewest modules. Max. span: 0, 5 and 10 modules.
  - Minimize the cost, i.e. the total height of supports
  - Placing supports in the lakes is forbidden.

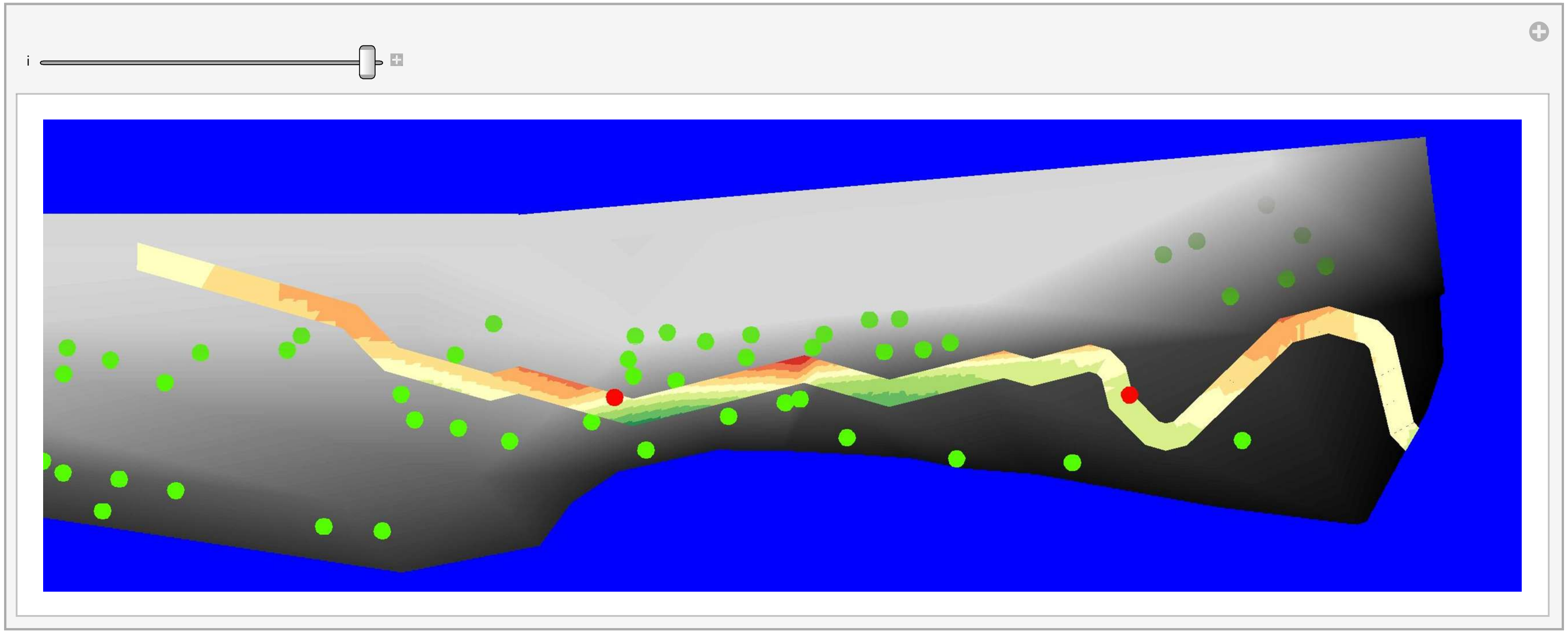


6. Optimization of Truss-Z with GPU. **Case-study 3**.

- “PKP Powiśle retrofitting” objective
  - Create TZ from P_0 to (P_1) which is not defined but given as area.
  - The length of TZ is derived from the elevation difference between P_0 and (P_1), which is 10.1 m (thus 101 TZ units)
  - Optimization objectives: minimal earthwork, tree-preservation (and optionally, *straightness*)



- 1: minimal earthwork, 2: no trees removed, 3: “balanced”, 4: straight.



7. Structural Optimization of Truss-Z Module

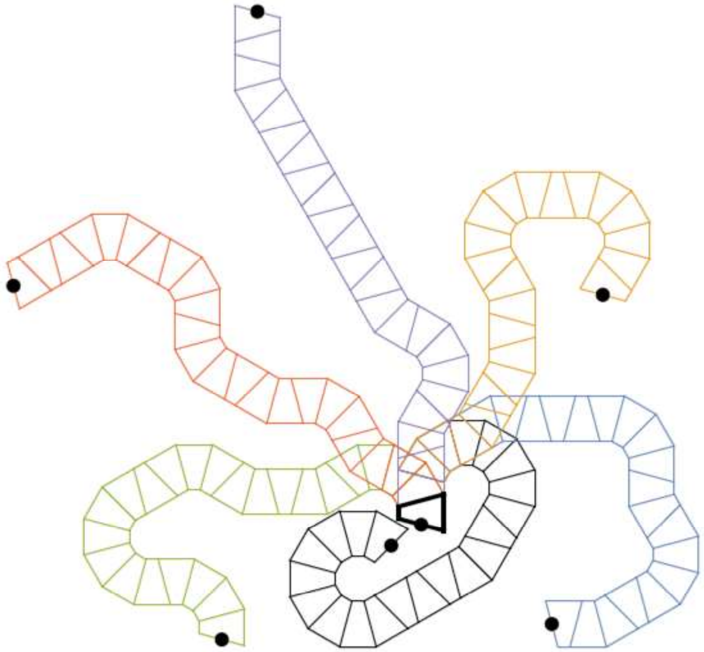
- TZ is modeled as a 3D frame structure
- TZM parameters
- How these parameters influence the entire TZ
- Example of an environment considered in the past
- The beams are modeled w/ finite element (FE) method as Euler-Bernoulli beams and circular hollow sections
- The maximum span for the universal TZM is 6 modules (1056 global essentially unique configurations to consider)
- The maximum von Mises equivalent stress under certain static design load is constrained from above.
- Sizing of the TZM beams is defined by their diameters (= optimization variables)
- Distribution of the static load

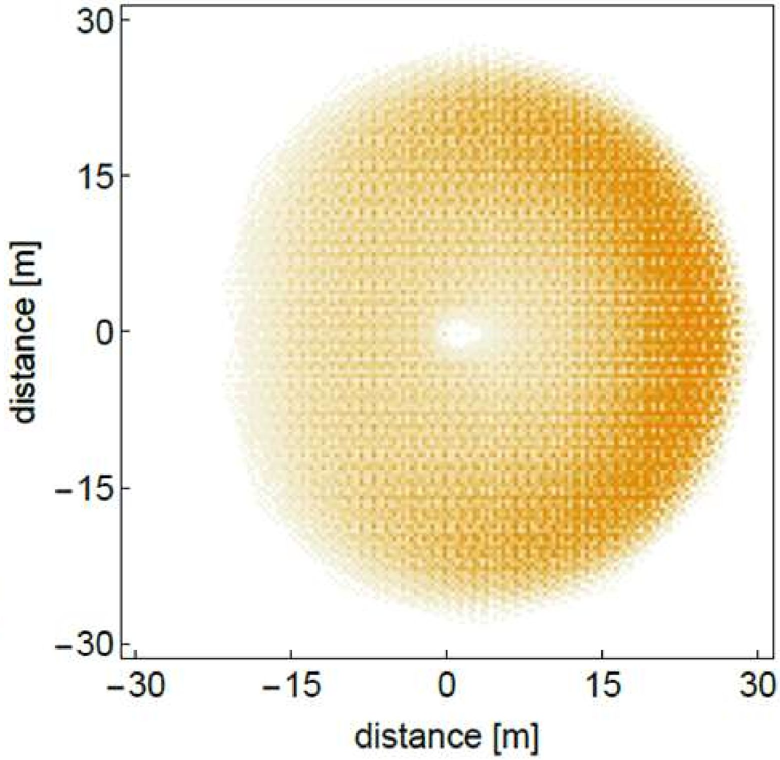
- Minimize the total mass of TZM, asses also compliance (typical aggregate measure of structural stiffness)
- Minimization of topology (diagonals) is performed for each of 16 possible orientations of the diagonal beams and for TZ lengths (from 2 to 7)
- Discrete-continuous optimization:

  Placement of the diagonals (*d*). Total of 16 different diagonals configurations

  Cross-sections of the module beams **x: x**= $\{\phi_1, \phi_2,...,\phi_{16}\}$

- Results
- Geometrical versatility: Geometrical in nature and represents the ability of the module to create free-form ramps of diverse shapes that possibly uniformly fill the spatial environment. On the right: bins 40x40 cm for up to 22 modules.





- index of dispersion of the point cloud (bin-count based):

$$I(\mathbf{x}) = \frac{s_p^2}{\bar{p}},$$

$$\bar{p} = \frac{1}{N}\sum_{i=1}^{N} p_i,$$

$$s_p^2 = \frac{1}{N-1}\sum_{i=1}^{N}(p_i - \bar{p})^2$$

- Structural measure: the minimum mass to length ratio of TZM that allows any six-module ramp branch, subjected to a given static design load, to satisfy an upper bound on the maximum effective stress.

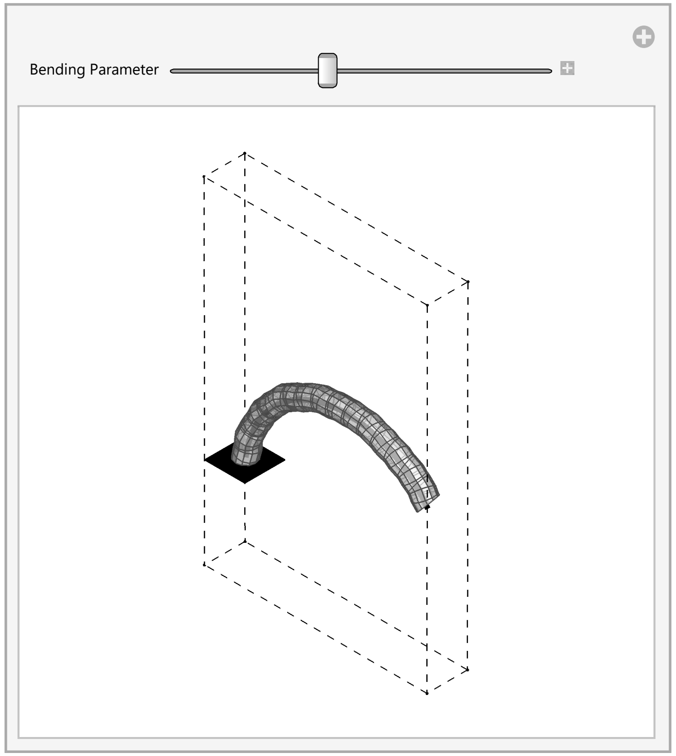
$$m(\mathbf{x},\mathbf{y}) = \frac{\pi d \sum_{i=1}^{16} (\phi_i - d) l_i(\mathbf{x})}{c_{\mathbf{xy}}}$$

subject to $\sigma_i^{\max}(\mathbf{x},\mathbf{y}) = \max_{s \in S} \max_{1 \leq n \leq \bar{s}} \sigma_{ins}^{\max}(\mathbf{x},\mathbf{y}) \leq 100\ \mathrm{MPa}$

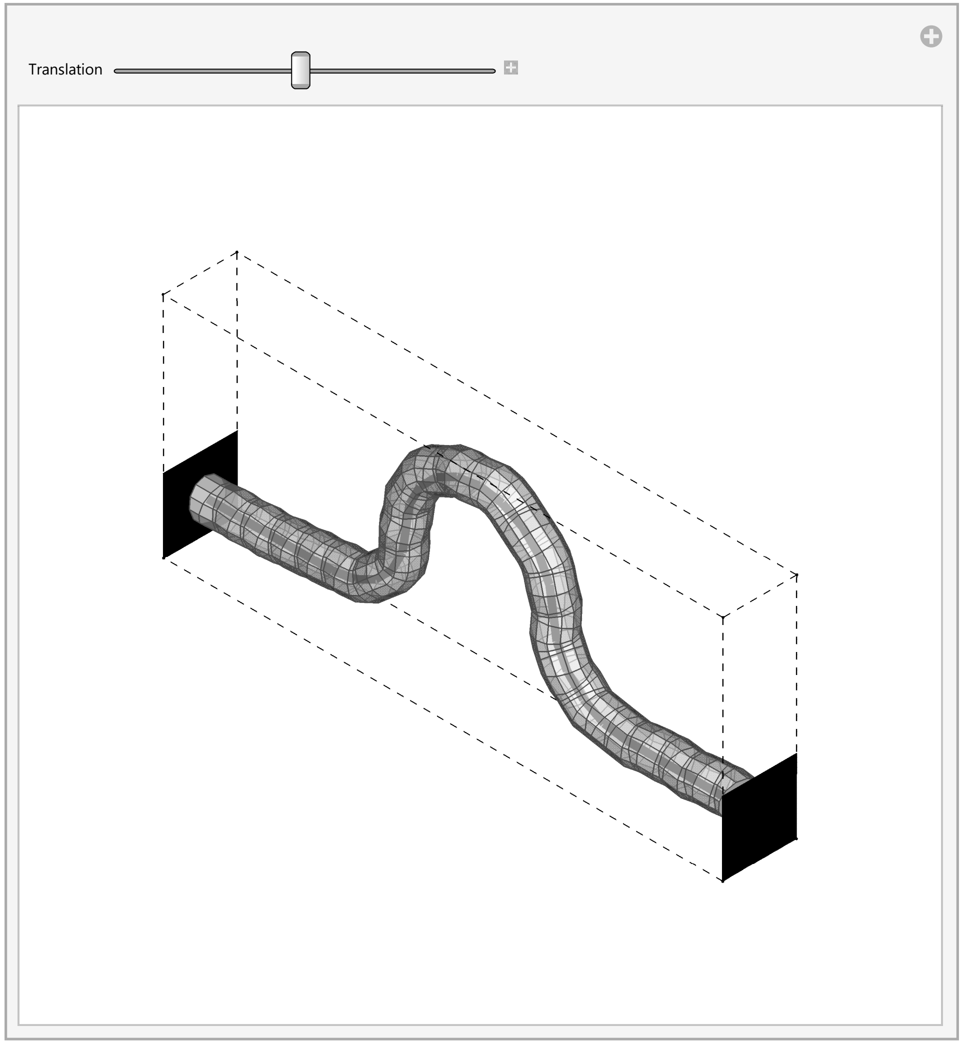
- Objective functions (geometrical and structural)
- Pareto front and the corresponding path in the design domain
- Optimal beam diameters along the Pareto front

8. **Arm-Z**: Three fundamental movements:

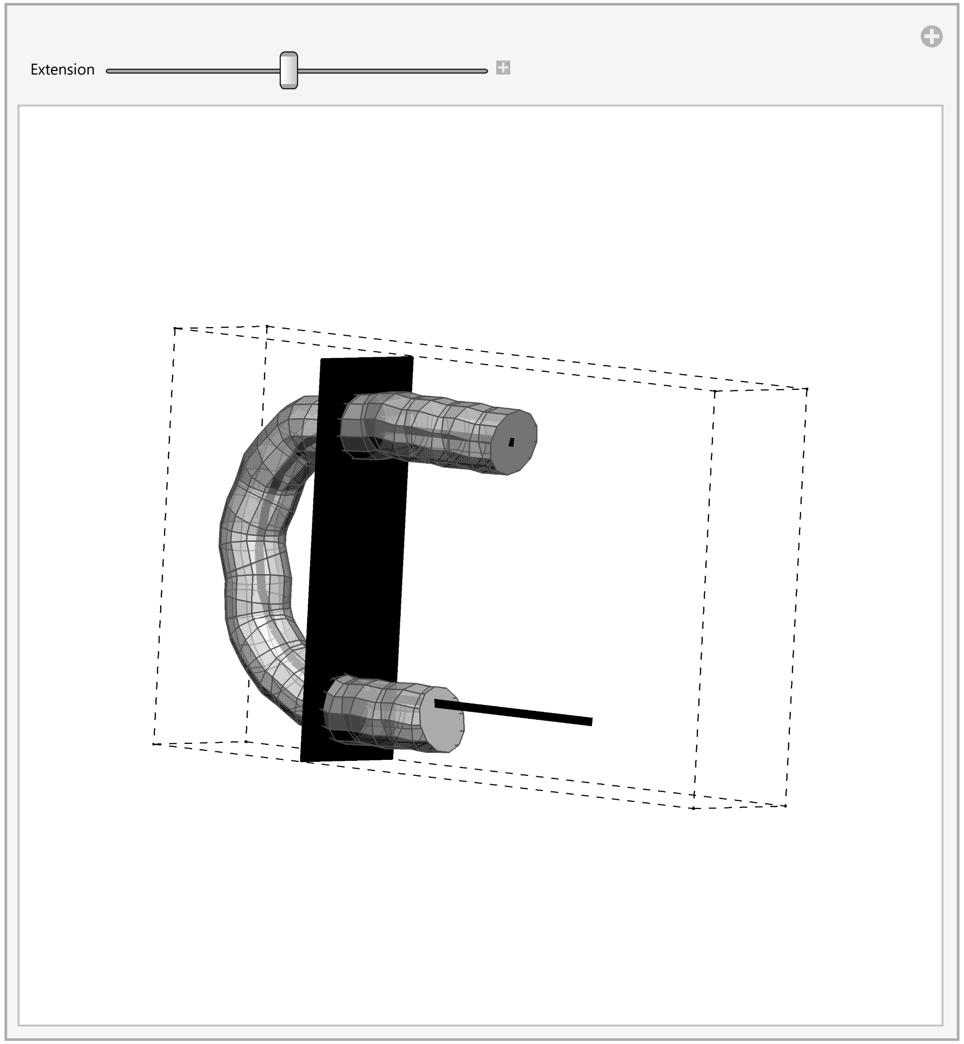
- Flexure



Translation

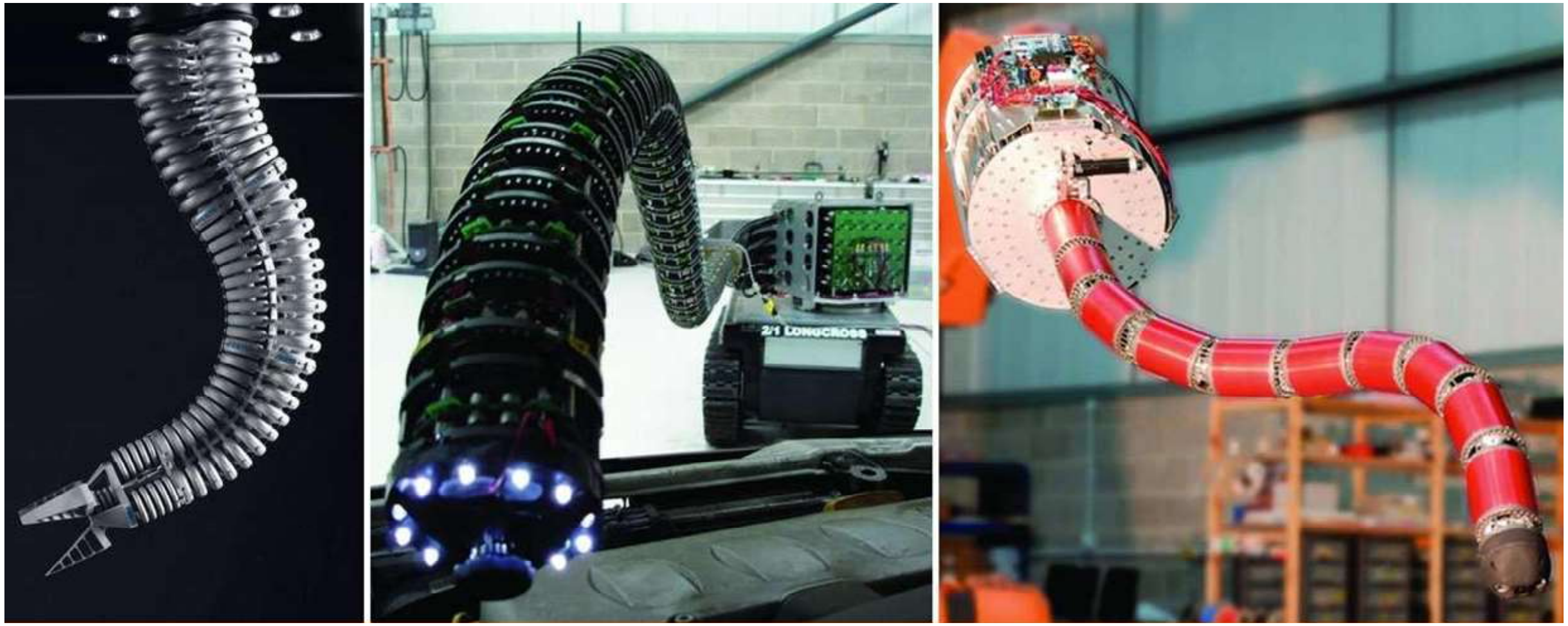


Extension



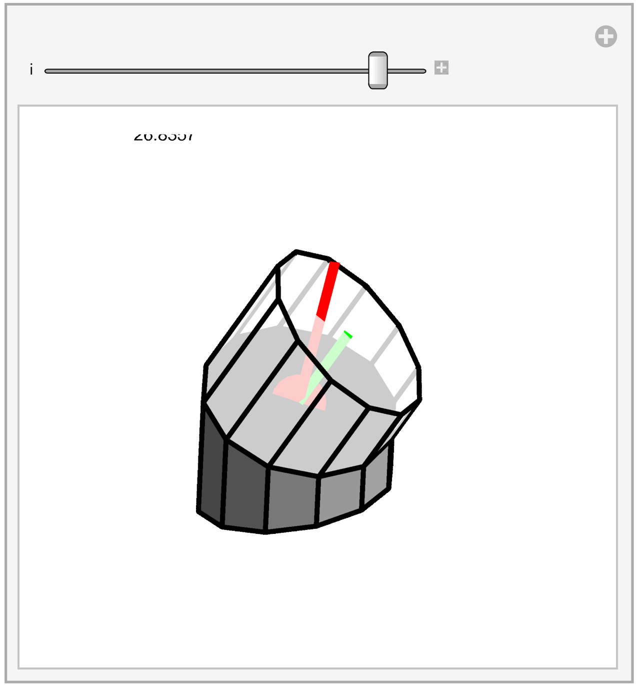
8. **Arm-Z** as a modular manipulator

- Examples of hyper-redundant manipulators (HRM)

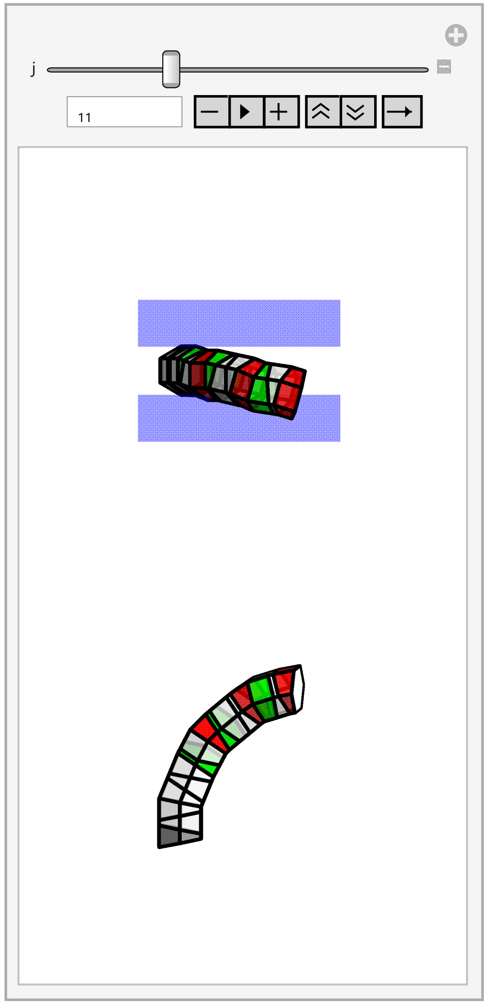


■ Dodecagonal 12-unit Arm-Z mock-up prototype

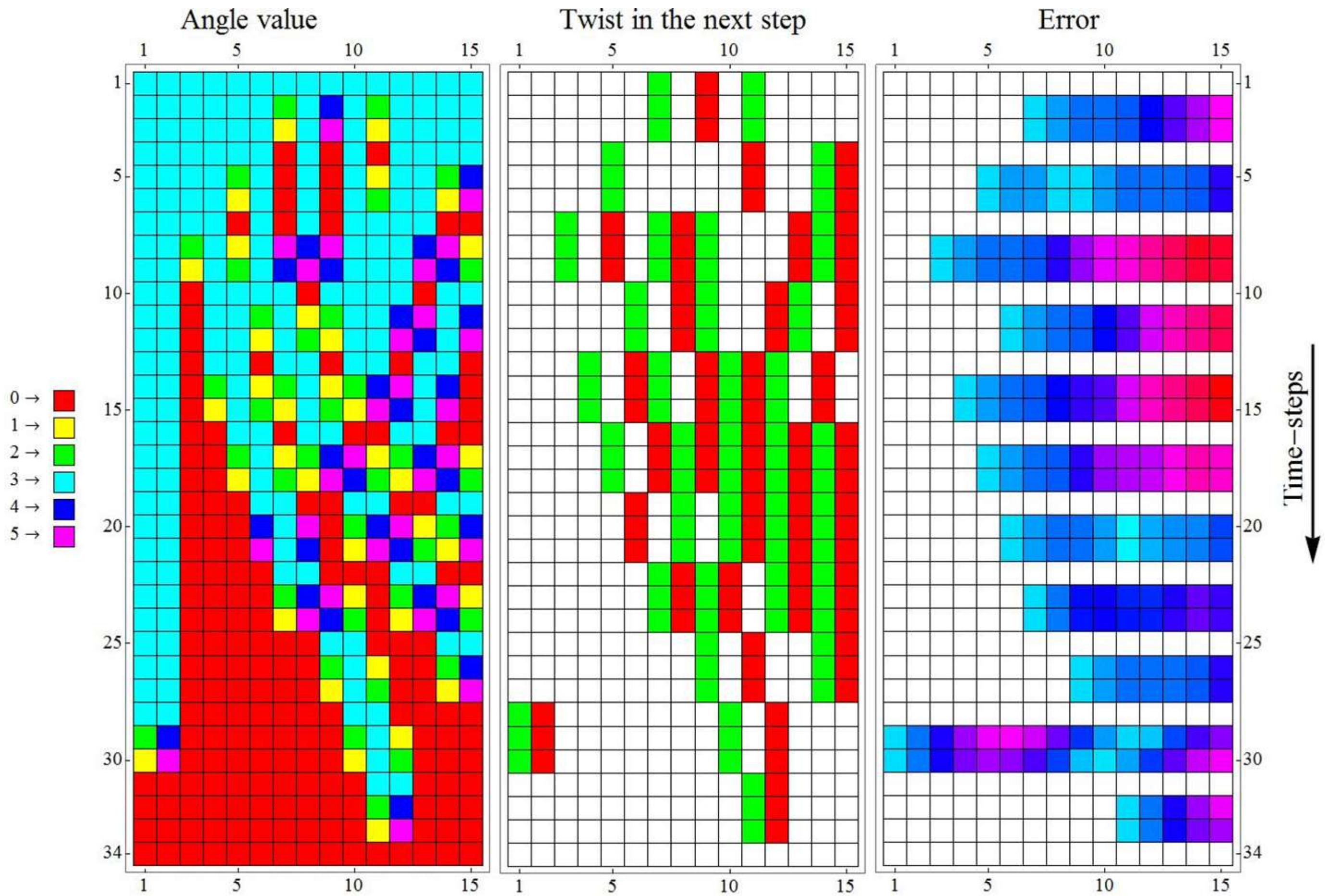
The control : dihedral rotations



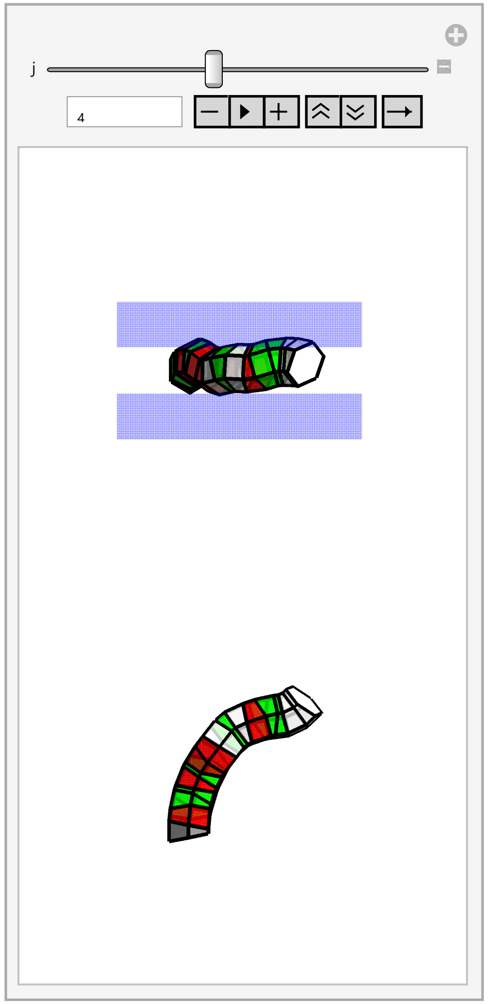
“Greedy” flexure of hexagonal



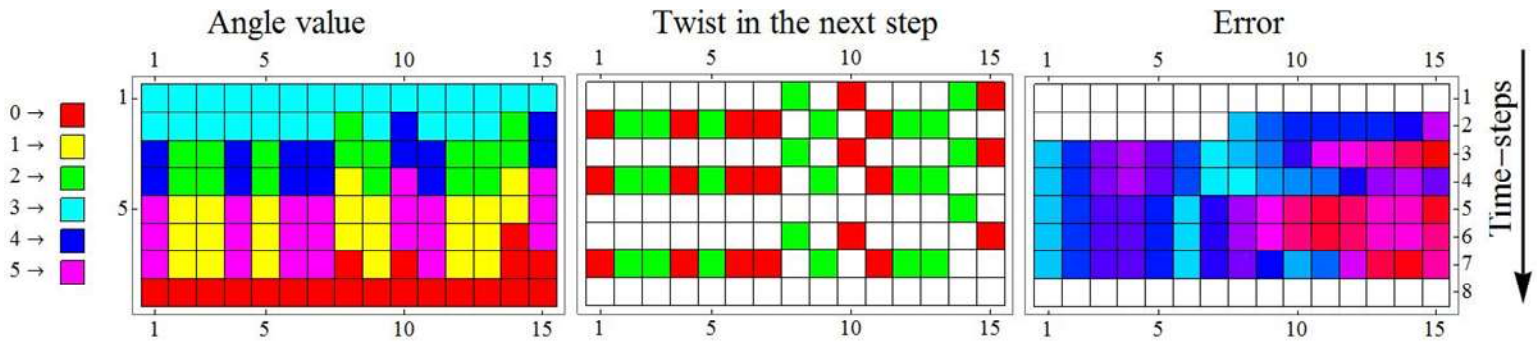
- Matrix-of-changes (MOC) for the “Greedy” flexure



- **Particle Swarm Optimization** is suitable due to the velocities encoded in real values

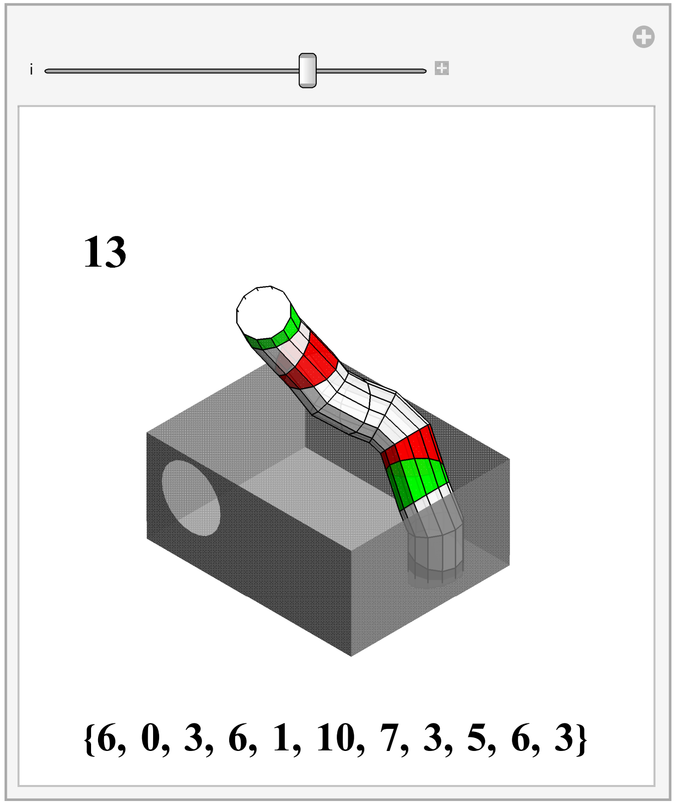


- MOC for PSO (only 8 steps)

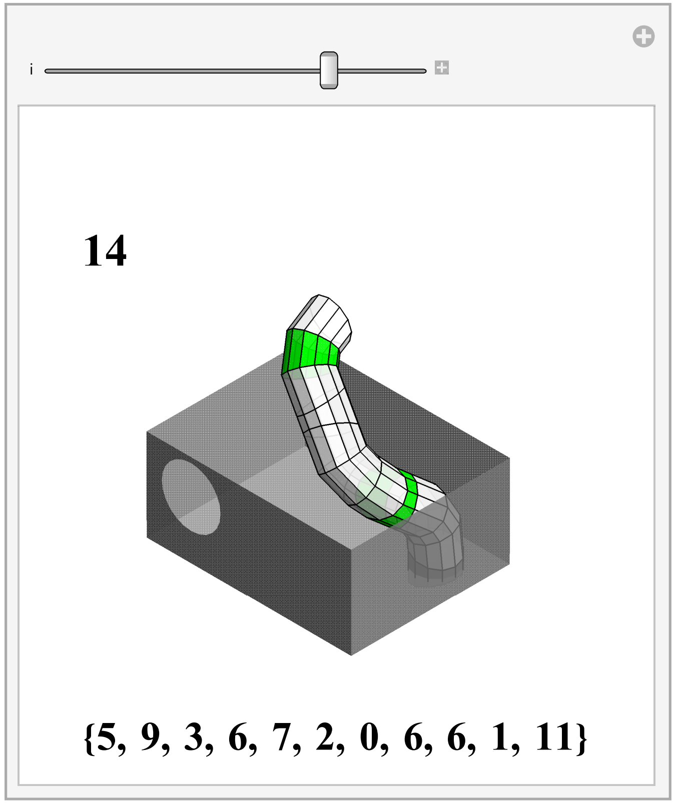


8. **Arm-Z**: realistic case-studies

- Known final state (PSO)



- Unknown final state (PSO)



Initializations

GPU Frames

Manual PZ frames

PZ examples

Arm - Z