Construction Automation with Bio-inspired Hierarchical Extremely Modular Systems

Ela Zawidzka, Machi Zawidzki

Abstract:

This paper presents the concept of hierarchical extremely modular systems (EMS). The biology-inspired nomenclature, genetic encoding, and operations for this class of structures are introduced and illustrated with various examples. Four mutation types are introduced and briefly analyzed. A relatively good convergence of the evolution strategy-based algorithm applied for optimization of EMS is shown.

Keywords: *evolution strategy, discrete structure encoding, extremely modular system, multi-branch structure*

1. Introduction

Construction automation deals with applying the principles of industrial automation to the construction sector or in the prefabrication of construction components.

Extremely Modular System (EMS for short) is a relatively new concept introduced a few years ago in [1]. It represents a new approach to the design of engineering structures and architectural objects where the assembly of congruent units allows for the creation of free-form shapes.

The main difference from the traditional modular systems used in engineering and building construction is the emphasis on the minimal diversity of types of modules, ideally—just one. This is why these systems are *extremely* modular.

These are five basic advantages of EMSs:
1) **Economical** – as they are suitable for mass fabrication, thus lowering the cost, so they can be broadly applied;
2) **Functional** – as they allow for reconfiguration, expansion, reduction;
3) **Robust** – since every module that fails can be easily replaced with an identical but functional one;
4) **Discrete** – as they are suitable for intelligent mathematical modeling, and their configurations can be subjected to discrete (multi-objective) optimization using efficient search algorithms;
5) **Uniform** – this feature is advantageous for rapid deployment and automated assembly.

Pipe-Z (PZ) is a more fundamental system introduced in [2]. Its purpose was to form spatial mathematical knots by assembly of one type of unit (PZM), as shown in Fig. 1.
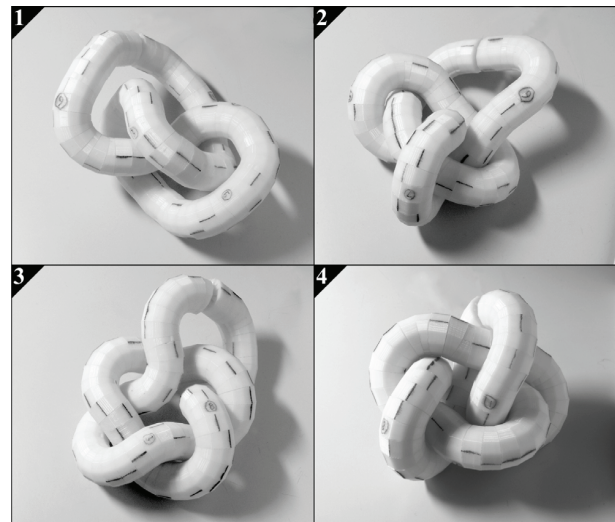

Figure 1. Physical models of mathematical prime knots constructed with extremely modular Pipe-Z: 1) Trefoil (3_1); 2) Figure-eight knot (4_1); 3) Cinquefoil (5_1); 4) 6_3 knot
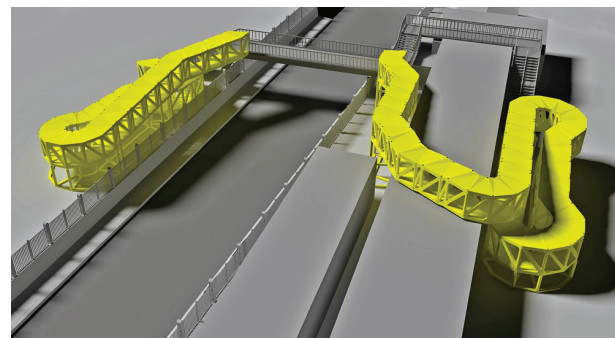
Figure 2. A computer rendering of an existing overpass retrofitted with two branches of Truss-Z, comprised of 47 modules on the left and 77 modules on the right

The shape of PZ is controlled by the relative twists of congruent modules in a sequence.

Truss-Z (TZ) is a skeletal-frame hybrid construction system introduced in [3] for creating free-form pedestrian ramps and ramp networks connecting any number of terminals in space. TZ is composed of four variations of a single basic module subjected to affine transformations (mirror reflection, rotation, and combination of both). Figure 2 shows an example of TZ.

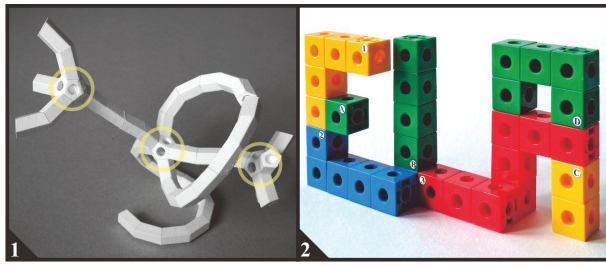**Figure 3.** 1) A physical model of the spatial multi-branch structure based on PZ. The bud is based on a truncated pentagonal prism. Three additional "branching buds" based on a dodecahedron (12-gon) are indicated by yellow circles. This system is called PZ_{12}* and is considered EMS-2. 2) An example of a spatial multi-branch quasi-EMS. This structure is built with "Space-Cube"—a system of toy blocks based on the simplest space-filling polyhedron—the cube. This system is called SC^3 for short
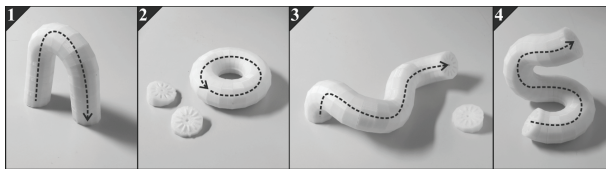

Figure 4. Examples of 3D PZ structures. The units have been added sequentially at various twists along the dotted arrow. 1) An arch; 2) a torus; 3&4) free-form pipes
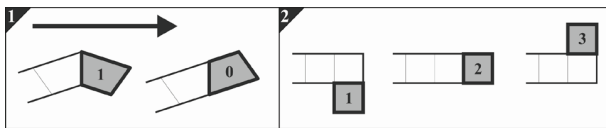

Figure 5. Planar projections of EMS (1) and qEMS (2)

Although originally Pipe-Z allows for the creation of single-branched structures, by the introduction of an additional "branching buds," it is also possible to construct multi-branched structures. Such a system is called EMS-2; an example based on PZ with an additional dodecahedron branching bud is shown in Figure 3.1.

2. The Nomenclature and Encoding

The basic elements for both EMS and qEMS are the *units*. They are added sequentially along the direction of the construction, as illustrated with PZ in Figure 4.

The core of the EMS concept is the shape of the unit, which allows the creation of various forms depending on its relative rotation. Its angular direction is relative to the directions of the *branches*, that is *stems* or *twigs.* For clarity all the following examples are reduced to 2D. Figure 5 shows examples of unit assembly for EMS and for the projection of SC^3 to 2D, called SC^2.

1) TZ is a strict EMS. By rotation, the basic unit can be added in two ways, determining the shape of the structure. 1 and 0 turn right and left along the
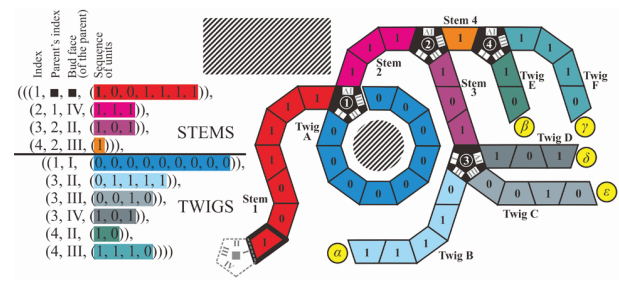

Figure 6. The nomenclature and encoding of EMS-2. On the left: the genotype (((1, ■, ■, (1, 0, 0, 1, 1, 1, 1)), (2, 1, IV, (1, 1, 1)), (3, 2, II, (1, 0, 1)), (4, 2, III, (1))), ((1, I, (0, 0, 0, 0, 0, 0, 0, 0, 0)), (3, II, (0, 1, 1, 1, 1)), (3, III, (0, 0, 1, 0)), (3, IV, (1, 0, 1)), (4, II, (1, 0)), (4, III, (1, 1, 1, 0)))) tabulated for clarity. On the right: the corresponding phenotype. The obstacles are indicated by hatched areas. The terminals are indicated by Greek letters. In this case, the buds have a form of pentagons (indicated by black). Buds and bud faces are indexed with Arabic and Roman numerals, respectively. The corresponding elements are shown in the same colors. Stems and twigs are shown in "reddish" and "bluish" colors, respectively. The first unit is indicated by a thick black outline. It starts from the "virtual" unit shown by a gray dotted line

direction of the structure, respectively. 2) SC^2: the basic unit of this qEMS can be added in three ways: at the right, front, and left face, as indicated by numbers 1, 2, and 3, respectively.

In genetic algorithms, all candidate solutions have sets of properties encoded in so-called genotypes, usually as binary strings, arrays, trees, lists, or matrices, which can be mutated and/or altered [6].

A phenotype is an actual EMS with its (physical) form and structure encoded in a given genotype. Obviously, there is a bijection (one-to-one correspondence) between the set of genotypes and phenotypes.

Figure 6 illustrates the nature-inspired naming for the components of an EMS phenotype and parts of the corresponding genotype, which is a simple nested list.

As illustrated in Figure 5, EMS-2 is comprised of <u>two</u> types of elements: the basic *unit* and *bud*. The *units* have two orientations: 0 and 1, and each *stem* ends with a pentagonal *bud*. For the consistency of notation, the initial unit is also attached to a "virtual" *bud*.

As mentioned in the Introduction, *plesiohedra* [4] are not considered as EMSs. The cube is the only regular solid that is also a *plesiohedron* [5]. Nevertheless, the multi-branched structures built with space-filling polyhedra can also be encoded in the same way. Figure 7 shows the genotype of the 2D projection (SC^2) of the cube-based qEMS structure (SC^3) shown in Figure 2.2.

As Figures 3 and 4 indicate, the hierarchical structures of EMS (and quasi-EMS) are defined as follows:

i. In general, the structure is constructed with *branches*.

ii. Branches are: stems and twigs.
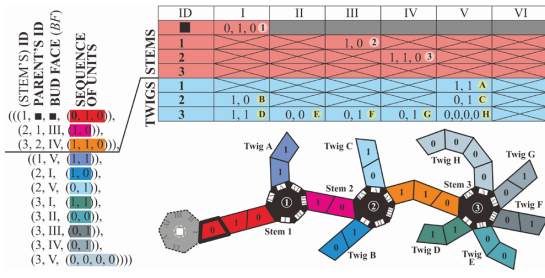
iii. *Stems* end with *buds. Twigs* do not have *buds.*

Figure 7. SC^2: the 2D projection of SC^3 structure shown in Figure 3.2. The genotype is shown on the left, and the corresponding phenotype is shown on the right. The respective corresponding elements are shown in the same colors. The last unit of each stem functions as a bud and is indicated by a diagonal hatch

iv. *Buds* can have any number of faces (*BF*s)

v. A *branch* can be connected only to a *bud*.

vi. A *stem* has at least one unit. In the case of a one-unit long *stem*, it becomes a *bud*.

vii. *Stems* must not form closed loops.

viii. The first *stem* is connected to a "virtual" *bud*.

ix. The first *stem* is also *rooted* to the initial terminal. There are no more *stems* directly attached to a terminal.

x. A terminal can only be reached by a *twig,* exclusive of the first *stem* and terminal.

Genotype of an ith Extremely Modular System G_i is encoded as a sequence of *branches.* The sequence of *branches* includes two sub-sequences of *stems* and *twigs:*

$$G_i : (S_i, T_i)$$
$$S_i : (s_1, s_2, \ldots, s_n)$$
$$T_i : (t_1, t_2, \ldots, t_m)$$

where S_i and T_i are the sequences of *n stems* and *m twigs*, respectively.

Moreover, s_j is a jth *stem* from the sequence of *stems* S_i encoded in the following way:

$$s_j : (j, p_j, f_j, u_j)$$

where: j is the index of the jth *stem*, p_j is its jth *parent*, the index of that is the stem to which the jth.

f_j is the index of the face of the *parent's bud (to which the jth stem is connected), and u is the sequence of units that* constitute the jth *stem.*

Units u depend on the type of EMS. E.g., for the systems shown in Figures 3.1 and 3.2: $u \in (0, 1)$ and $u \in (1, 2, 3)$, respectively. *Buds* also depend on the type of EMS. E.g., Truss-Z and SC^3 do not have additional units for *buds*.

In Truss-Z, buds are constructed by reflecting the last unit in the stem sequence across the longer base of the trapezoidal unit.

In SC^3, the last unit in a *stem* serves as a branching *bud*. On the other hand, the *buds* for EMS-2 systems can have any shape. In the examples shown in Figures: 3.1, 6 and 8 & 9 they are: dodecahedron, pentagon and heptagon, respectively.

Since the first *stem* does not have a *parent*, p_1, and f_1 are dummies (blank).

The kth *twig* from the sequence T_i is defined in the following way:

$$t_k : (p_k, f_k, u_k)$$

where p_k is the *parenting stem* to which this *twig* is connected,

f_k, is the index of the face of the *parent's bud* (to which the kth *twig* is connected), u_k is the sequence of units that constitute the kth twig.

There is a full analogy between the list of *twigs* and *stems*.

3. Operators on the EMS Genotype

There are seven operators collected in Table 1 that allow the construction of any EMS or the transformation of any EMS_A to any other EMS_B.

In addition, four operators for the population-based algorithm are defined and listed below.

Random genotype generator

$$rG\,[(s_{min}, s_{min}), (t_{min}, t_{max}), (u_{min}, u_{max})]$$

where s is the number of *stems*, t is the number of *twigs*, and u is the number of *units* to be generated in all *branches*.

Fix genotype

$$F\,[G_k, s, t, l]$$

where G_k, is the given k^{th} genotype,
s is the desired number of *stems*,
t is the desired number of *twigs*,
l is the number of units generated randomly in additional branches, if applicable.

Tabulate genotype

This function transcribes genotype G_i into matrix $T[G_i]$ with respect to the *buds*. See Figure 8 for an illustrative example.

Compare (tabulated) genotypes

$$C[G_j, G_k] = \Delta(T * [G_j], T * [G_k])$$

Table 1. Steps for creating an Extremely Modular System or its transformation

Start	Transformation	Operator
General structure	↳	
↓	Add a *stem*	$aS\,[G_i, (i_i, BF_i, u_i)]$
	Add *twigs*	$aT\,[G_i, ((p_1, BF_1, u_1),(p_2, BF_2, u_2),...,(p_k, BF_k, u_k))]$
	Remove *branches*	$rB\,[G_i, ((p_1, BF_1), (p_2, BF_2), ...,(p_i, BF_i))]$
Substructure @ buds	↩	
	↳	
↓	Displace *branches*	$dB[G_i, (((p_i, BF_i), (p_j, BF_j)),...)]$
Units @ branches	↩	
	↳	
↓	Add units @ branches	$aU\,[G_i,((p_1,BF_1,(v_1^1,l_1^1),...,(v_k^1,l_k^1)),...,(p_h,BF_h(v_1^l,l_1^l),...,(v_k^l,l_k^l)))]$
	Remove units @ branches	$rU\,[G_i,((p_1,BF_1,(l_1^1,...,l_k^1),...,(p_h,BF_h(l_1^l,...,l_k^l)))]$
	Invert units @ branches	$iU\,[G_i, ((p_1,BF_1,(l_1^1,...,l_k^1),...,(p_h,BF_h(l_1^l,...,l_k^l)))]$
End	↩	

Figure 8. From the left: the genotype; top right: the corresponding tabulated genotype; on the bottom right: the corresponding phenotype
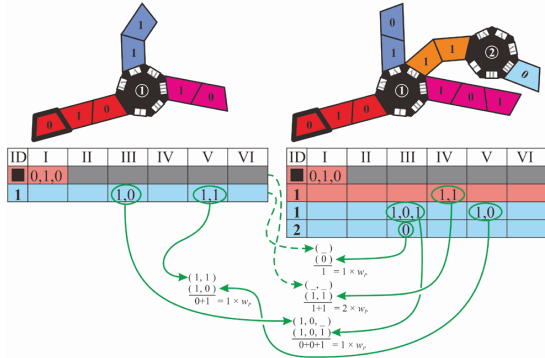

Figure 9. Calculation of the difference between two exemplary genotypes. The differences for all corresponding branches are summed up and multiplied by the penalty weight w_P. Missing corresponding branches are indicated by dashed arrows. The difference between the two empty lists is 0

where $T^*[G_j]$ is the normalized and tabulated genotype G_j and $T^*[G_k]$ is the normalized and tabulated genotype G_k.

Normalization here means that $T[G_j]$ and $T[G_k]$ have the same general structure. For example, if the dimensions of these genotypes differ, dummy elements are added. Moreover, $C[G_j, G_k] \geq 0$.

Δ sums up the differences between the bit-strings of respective *branches* of the compared genotypes. Figure 9 illustrates the way how Δ is calculated.

Finally, four types of mutation and their combination have been defined as follows:

Displace-branch-mutation

$$MdB\,[g_k, m_i]$$

where g_k is the kth genotype, m_i is the normalized intensity of mutation, so that for 0 there is no displacement and for 1 all branches are displaced.

Add-unit-mutation

$$MaU\,[g_k, m_i]$$

in this case, m_i is normalized so that for 0, no units are added, and for 1, the number of units is doubled. Loci for the added units are distributed randomly among the *branches*. The values of the added units are random integers: 0 or 1.

Remove-unit-mutation

$$MrU\,[g_k, m_i]$$

Unlike *stems*, *twigs* can have zero length. Here, m_i is normalized so that for 0, no unit is removed, and for 1, all units but one randomly selected unit per *stem* are removed. The loci for units to be removed are distributed randomly among the *branches*.

Invert-unit-mutation

$$MiU\,[g_k, m_i]$$

where m_i is normalized, so that for 0 there is no change to any unit, and for 1 the values of all units are inverted. The loci for units to be inverted are distributed randomly among the *branches*.

Multi-mutation (combines all types of mutations defined previously)

$$M[g_k, m_i, (w_{db}, w_{aU}, w_{rU}, w_{iU})]$$

where m_i is the normalized intensity of mutation,

M randomly selects the mutation type according to parameters w_{db}, w_{aU}, w_{rU}, and w_{iU}, which are the normalized weights (from 0 to 1) for corresponding mutations: *MdB*, *MaU*, *MrU*, and *MiU*.

4. Minimization by Evolution Strategy

Evolution Strategy is a classic nature-inspired heuristic method introduced in [7].

Unlike other evolutionary algorithms, it does not employ recombination but only intensive mutation. For a review of the literature on this method, see [8].

This experiment is a two-objective <u>minimization</u> defined as follows:
1) Create a layout of a multi-branch Truss-Z (MTZ).
2) There are six terminals at given locations: the initial terminal (T_S), and five aligned terminals $T_1...T_5$, as shown in Figure 15.
3) The (initial) direction of the first unit is given at T_S.
4) The number of units is to be <u>minimal</u>.
5) The *reaching error* (rE) is to be <u>minimal</u> (explained below).

Reaching error (rE) is the sum of distances between each terminal and respective *twig* tips, as illustrated in Figure 10.

The cost function CF for a given i^{th} multi-branch Truss-Z (MTZ_i) with m number of destination terminals is calculated as follows:

$$CF(MTZ_i) = n + w_p \times rE = \int_{i=1}^{m} \varepsilon_i$$

where n is the total number of units in MTZ, ε_i is the distance between the ith terminal and corresponding *twig* tip, w_P is the penalizing weight adjusted by trial-and-error; for small values of w_P algorithm gets stuck in local minima and as a result, solutions do not improve.
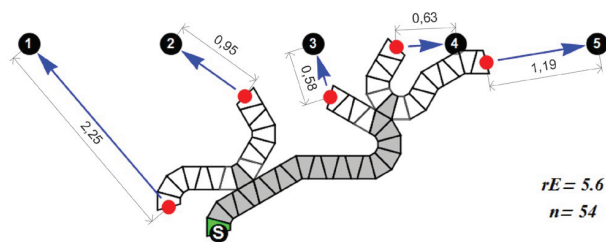
Figure 10. In this case, reaching error rE = 2.25 + 0.95 + 0.58 + 0.63 + 1.19 = 5.6. The number of units n = 54. Stems with branching units are indicated in gray. Twigs are white
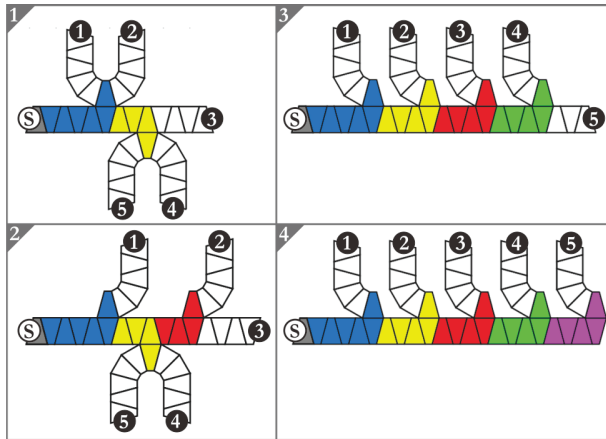

Figure 11. Sub-figures 1, 2, 3, and 4 show allowable MTZs connecting five terminals with: 2, 3, 4, and 5 stems, respectively

The initial population is generated randomly using *rG* operator described above. However, it is rational to start from the most reasonable initial population by using certain observations. E.g., the number of *twigs* must be exactly 5. This is because only twigs can reach the terminals, and there are five of them. However, with the number of *stems* it is not so obvious. Figure 11 shows four different allowable MTZ configurations connecting five terminals with the initial terminal T_S.

Therefore, the number of stems in every candidate solution must be within the range [2, 5]. The initial population of 200 MTZs has been generated with the following parameters:

$$rG\,[(s_{min}, s_{max}), (t_{min}, t_{max}), (u_{min}, u_{max})]$$

the number of *stems*: $s_{min} = 2$, $s_{max} = 5$;
the number of *twigs*: $t_{min} = t_{max} = 5$;
the number of units in each *branch*: $u_{min} = 5$, $u_{max} = 15$.

Several trials at various combinations of parameters have been performed. Each experiment has been terminated after 100 iterations. Figure 12 shows the best phenotypes in the final trial.

As Figure 12 indicates, in the beginning, several individuals were infeasible due to self-intersection. Nevertheless, relatively soon, this problem disappeared, as gradually the feasible offspring were produced by infeasible parents. Figure 13 shows the final solution and its three-dimensional interpretation.
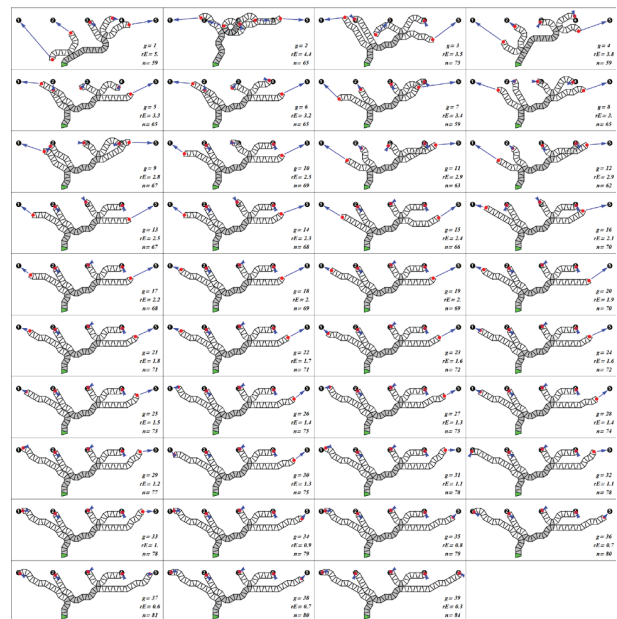

Figure 12. The final trial of the Evolution Strategy-based experiment. For each phenotype, the generation number (g), the reaching error (rE), and the number of units (n) are shown in the bottom right corner. There has been no further improvement after 39 generations. Stems with branching units are indicated in gray. Twigs are white
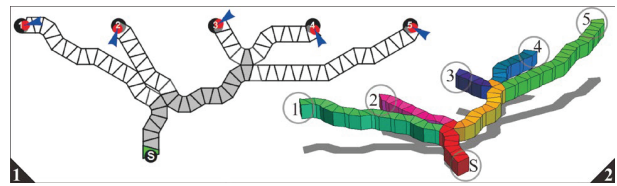

Figure 13. 1) The best MTZ layout produced by Evolution Strategy. 2) The three-dimensional MTZ based on this layout

5. Conclusion

The concept of hierarchical structures has been outlined in the context of Extremely Modular Systems. The biology-inspired nomenclature, genetic encoding, and operations for this class of structures have been explained and illustrated.

Although the evolution strategy-based algorithm presented here has not been optimized for efficiency, it shows relatively good convergence. For more information on this approach, see [9].

AUTHORS

Ela Zawidzka[*] – ORCID: 0000-0003-1243-9355, Department of Intelligent Technologies, Institute of Fundamental Technological Problems of the Polish Academy of Sciences, Poland, e-mail: zawidzka@ippt.pan.pl.

Machi Zawidzki – ORCID: 0000-0001-8695-4400, Łukasiewicz Research Network, Industrial Research

Institute for Automation and Measurements PIAP, Poland, e-mail: zawidzki@piap.lukasiewicz.gov.pl.

*Corresponding author

References

[1] M. Zawidzki, *Discrete optimization in architecture: Extremely modular systems*. Springer: Singapore, 2017.

[2] M. Zawidzki and K. Nishinari, "Modular Pipe-Z System for Three-Dimensional Knots," *Journal for Geometry and Graphics*, vol. 17, no. 1, 2013, pp. 081-087.

[3] M. Zawidzki, "Creating Organic 3-Dimensional Structures for Pedestrian Traffic with Reconfigurable Modular 'Truss-Z' System," *International Journal of Design & Nature and Ecodynamics*, vol. 8, no. 1, 2013, pp. 61–87. DOI: https://doi.org/10.2495/DNE-V8-N1-61-87.

[4] B. Grünbaum and G.C. Shephard, "Tilings With Congruent Tiles." *Bull. Amer. Math. Soc.*, 3, 1980, pp. 951–973.

[5] M. Gardner, *The Sixth Book of Mathematical Games from Scientific American*. University of Chicago Press: Chicago, IL, 1984.

[6] A.E. Eiben, J.E. Smith, eds., *Evolutionary Algorithms, in F. Neri, C. Cotta, and P. Moscato: Handbook of Memetic Algorithms, Studies in Computational Intelligence*, vol. 379, Springer, 2012, pp. 9–27. DOI: https://doi.org/10.1007/978-3-642-23247-3_2.

[7] I. Rechenberg, "Evolutionsstrategie: Optimierung Technischer Systeme Nach Prinzipien Der Biologischen Evolution," Ph.D. thesis, Stuttgart, 1973 (in German).

[8] T. Bäck, F. Hoffmeister, H-P. Schwefel, "A Survey of Evolution Strategies, in R. K. Belew and L. B. Booker (Eds.)," *Proceedings of the Fourth International Conference on Genetic Algorithms*. University of California, San Diego, Morgan Kaufmann, 1991.

[9] M. Zawidzki, "Optimization of Multi-Branch Truss-Z Based on Evolution Strategy," *Advances in Engineering Software*, vol. 100, 2016, pp. 113–125. DOI: https://doi.org/10.1016/j.advengsoft.2016.07.015