Blucher Mechanical Engineering Proceedings
May 2014, vol. 1 , num. 1
www.proceedings.blucher.com.br/evento/10wccm

10th World Congress on Computational Mechanics
8-13 July 2012 • São Paulo • Brazil

APPLICATION OF EVOLUTIONARY ALGORITHM FOR OPTIMIZATION OF THE LAYOUT OF A MULTI-BRANCHED NETWORK CONSTRUCTED WITH TRUSS-Z SYSTEM

M. Zawidzki[1] , K. Nishinari[1]

[1]Research Center for Advanced Science and Technology, University of Tokyo, Japan
(zawidzki@gmail.com)

Abstract. *This paper concerns with the optimization of a planar layout of the multi-branched network created with Truss-Z (TZ) linking six terminals in an environment with two rectangular obstacles. This multi-objective optimization problem has two criteria: the number of modules to be the smallest (economical optimization) and none of the modules should collide with any other objects (obstacles or other modules of the network). Although the TZ structures are three-dimensional, in this paper, the problem of planar layout of the multi-branched network is considered.*
At first, a simple and robust backtracking-based algorithm is demonstrated. Since it finds the first allowable solution, that is a local optimum, the configuration is generated in a short time. Such a solution, however, is most likely not the best globally. Next, an evolutionary algorithm is implemented. The encoding of a planar multi- branched network of TZ, the selection method, the cost function and the genetic operations are introduced. A number of trials have been performed, and the results are briefly discussed and interpreted.

Keywords: *Truss-Z, modular ramp system, layout optimization, multi-branched network.*

1. INTRODUCTION

Truss-Z (TZ) is a concept of a modular skeletal system for creating free-form transportation links and networks among any number of terminals in space [1,2]. TZ is intended for pedestrians, especially ones with strollers or carts, wheelchairs, cyclists etc. In other words, for people who have difficulties using regular stairs. The underlying idea of this system is to create structurally sound provisional or permanent structures at the minimal number of types of modular elements. The TZ structures are composed of only two units – R and L which are mirror reflection of each other. By rotation, they can be assembled in two additional ways (R_2 – rotated R and L_2 – rotated L), effectively giving effectively there are four possible types of units. Some examples are shown in figure 1.

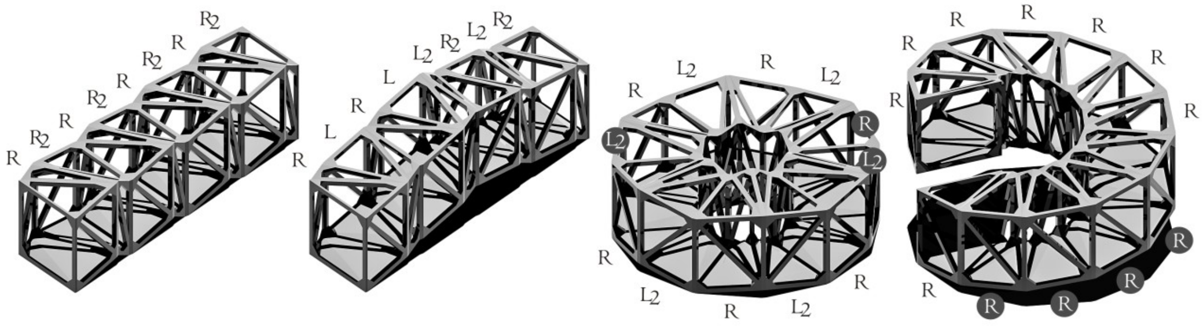


Figure 1. Some basic examples of single-branch TZ structures. From the left: “straight and flat” with 8 units, “straight up & down” (8), a flat ring (12), and a spiral (12).

Most importantly, TZ allows automated creation of optimal structural linkages for given terminals and obstacles [3]. The optimization criteria can be the minimal number of units, in case of multiple branches – minimal network distance etc. Efficient algorithms for two-dimensional single-branch TZ linkages in constrained environment have been demonstrated [4,5]. It is also possible to create a multi-branched TZ networks as shown in figure 2.

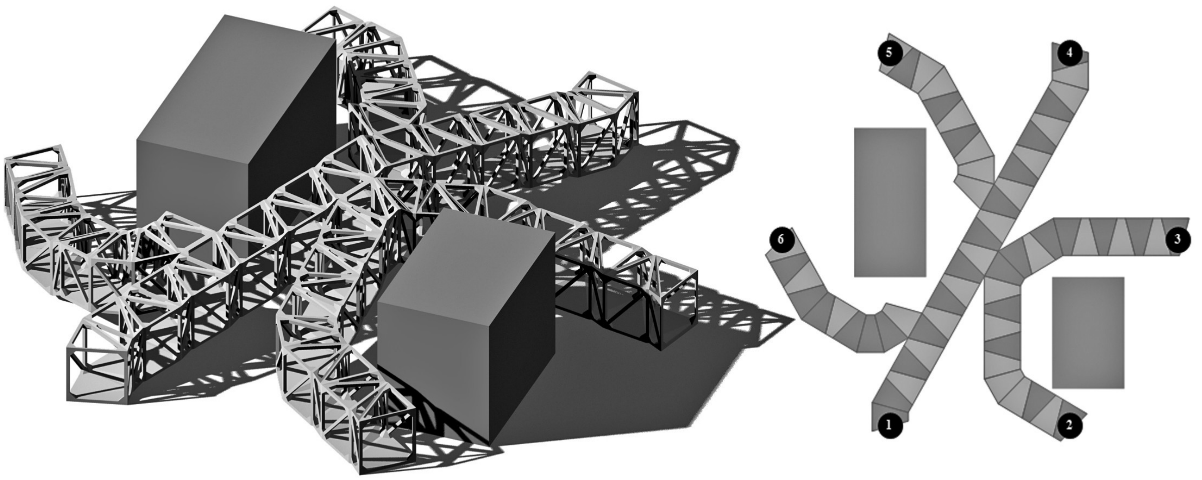


Figure 2. An example of a six-terminal TZ network. From the left: the isometric view of a 3D TZ structure composed of four branches and the network layout.

This paper focuses on efficient methods of laying out a TZ network starting from the terminal 1 and connecting the remaining five terminals without colliding with itself and two obstacles as shown in figure 2. The domain of the problem is reduced to 2D, thus instead of four variations of a unit there are only two: *r* (right) corresponding to R and R_2, and *l* (left) corresponding to L and L_2. Since there is an additional position for a branching unit *B*, the networks are laid out with a single trapezoidal unit that can be connected to other units in three different ways, as shown in figure 3.

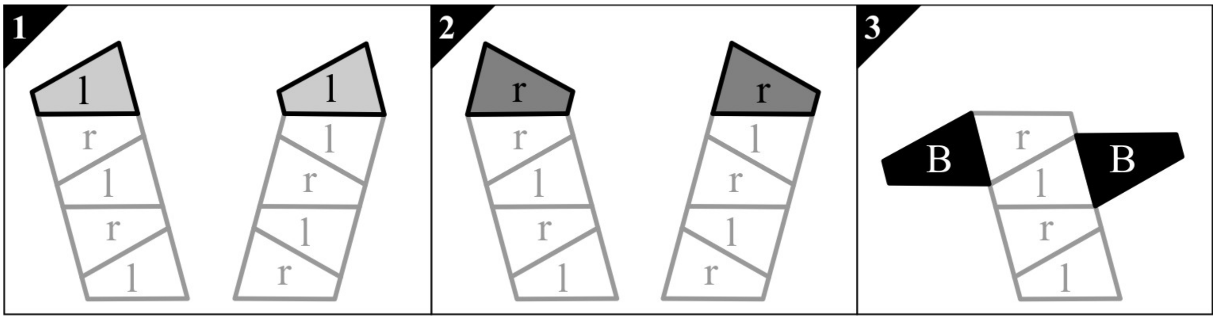


Figure 3. Three possible ways of attaching a new unit: depending on the type of the previous unit the path runs straight or turns. The branching unit B can be attached to the longer base of a trapezoid only.

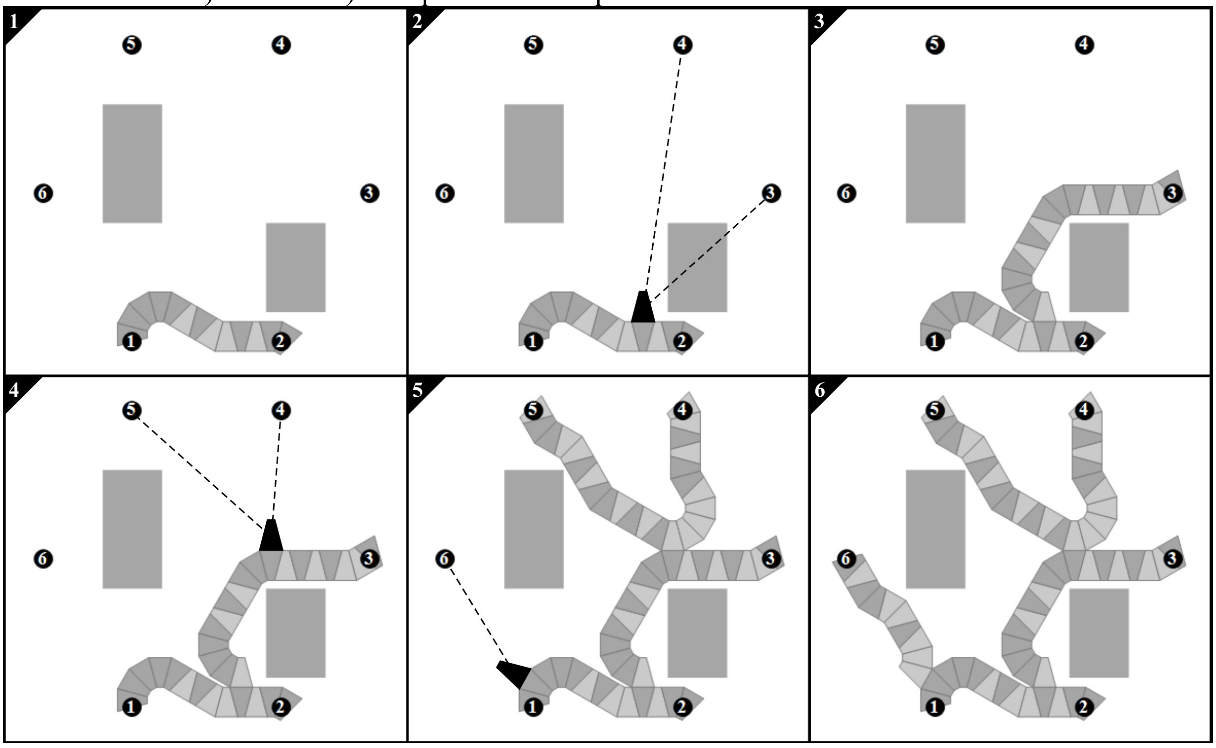
2. BACKTRACKING

At first a backtracking-based algorithm was implemented. The branches of the network are produced consecutively. The procedure of linking any two terminals (from Start to Goal) is defined as follows:

1. Start from the given terminal (Start)
2. At each step a unit (r or l) whose centroid is closer to the Goal is chosen.
3. If any point of the unit collides with any object, the procedure steps back, switches the last unit (r→l or l→r), and continues as in 1).
4. If any point of the unit still collides with any object, the procedures steps further back, switches the second last unit and continues as in 2) until reaching the Goal.

The adaptation of this procedure for the case with six terminals is shown in table 1.

Table 1. 1) The initial branch starts from the terminal 1 and proceeds to the closest terminal (2). 2) An available unit, that is a unit from which it is possible to construct a branching B (see figure 3) towards the next two consecutive terminals, is selected so that the sum of the distances to them is minimal. 3) From that unit branches are constructed, if possible. 4) As in 2. 5) As in 3. 6) The procedure stops when the last terminal is reached.



The procedure shown above is robust and in most cases produces quite satisfactory results. However, it is almost certain that better solutions exist. Since this method relays on partial solutions that are local minima, this method can not reach the globally optimal solution. Since the system is composed of discrete units, the number of possible combinations is always finite. However, since in the general (3D) case there are four variations of a unit (R, R_2, L, L_2), the number of all configurations for a single run of the truss grows exponentially as 4^n, where n is the number of units. For a planar case it grows as 2^n. The number of possible reconfigurations is additionally upsurged by the presence of multiple branches. Thus since the search space becomes so enormous, it is natural to apply heuristic method in order to investigate whether a better or alternative configurations exist.

2. EVOLUTIONARY ALGORITHM

The implementation of a classic evolutionary algorithm (EA) requires the following:

1. encoding of the solution into a genotype, that is a list of symbols,
2. selection method,
3. operation of crossover,
4. operation of mutation,

5. formulation of the fitness function, which should consistently give higher (for maximization) or lower (for minimization – as in this example) values for better candidates.

2.1. The genotype

In order to perform genetic operations the genotype should be a one-dimensional string. However, presently, the solutions are encoded as nested lists as explained in figure 4.

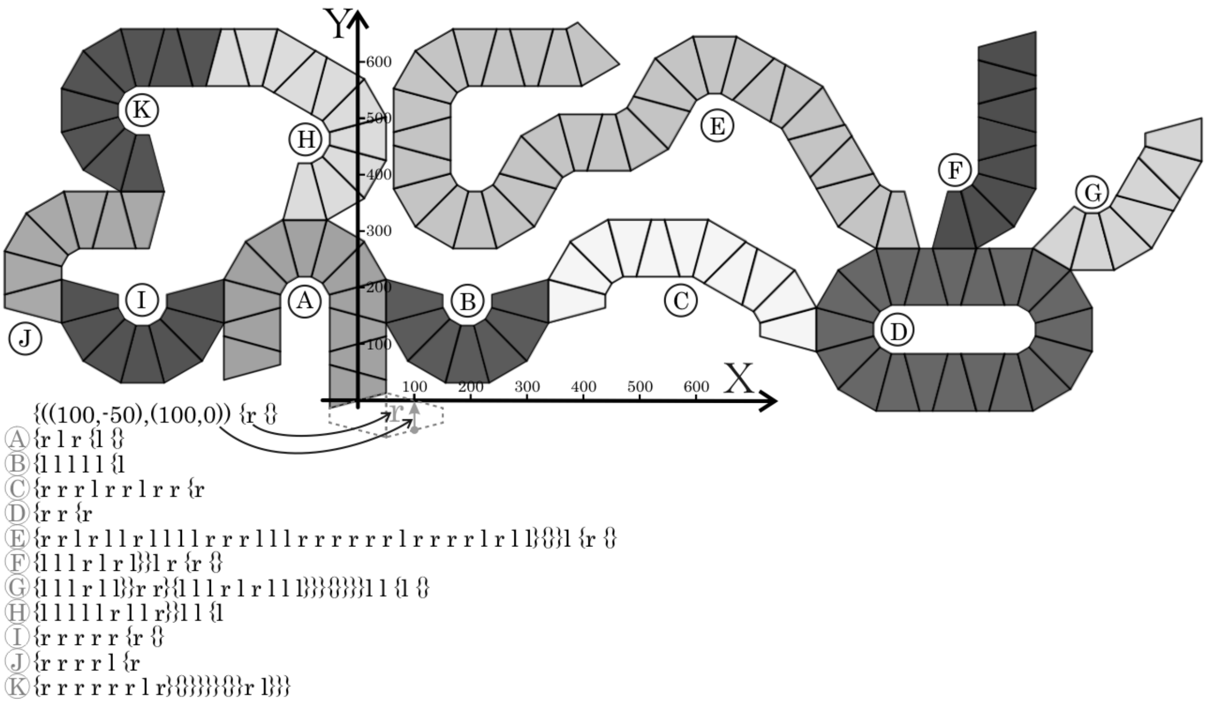


Figure 4. A genotype: {r {} {r l r {l {} {l l l l l {l{r r r l r r l r r {r{r r {r {r r l r l l r l l l l r r r l l l r r r r r r l r r r r l r l l} {}} l {r {} {l l l r l r l}} l r {r {} {l l l r l l}} r r} {l l l r l r l l l}}} {}}}} l l{l {} {l l l l l r l l r}} l l {l{r r r r r {r {} {r r r r l {r{r r r r r r l r} {}}}}} {}} r l}}, and the corresponding complex network layout. From each branching unit two wings can be constructed: on the left and right. An empty list {} stands for the absence of a wing.

2.2. Selection

Selection is the stage when parent solutions are chosen from a population for further breeding. The most common selection methods are the Tournament (TS) and Roulette (RS), also called fitness proportionate selection. TS is based on random selection of two candidates and choosing the better one for a parent. In RS, the parent solutions are randomly selected from the population with the probability proportional to the quality of a solution. It can not be generalized whether one of the methods is always better than another. The decision should be made for an individual problem.

2.3. Crossover

The core of EA is the assumption that the desired qualities of the individuals can be carried out and augmented by selective breeding throughout a reproduction process. The most commonly used methods are one-point (OPX) and uniform (UX) crossover. With OPX a single crossover point on both parents' genotypes is selected. All data beyond that point in either one is swapped between the two parents. With UX, the genes of one parent at selected points of the genotype (called *loci*) are replaced with the genes at the same loci of the other parent. In this paper the number and positions of loci are random. As with selection method, the most suitable type of crossover depends on a specific problem.

2.4. Mutation

As crossover maintains the general direction of the improvement towards the global optimal solution, the operation of mutation prevents the population from degenerating, that is excessive decrease of diversity of the individuals. In other words, it prevents the process from “getting stuck” in local optima. In this case there two types of mutation were combined:

1. Unit switch mutation (USM), where randomly selected genes are assigned the opposite value ($r \rightarrow l$ or $l \rightarrow r$). USM is controlled by two parameters – m_P which controls the probability of the occurrence of a mutation, and mutation intensity (m_I) which determines the number of genes to be mutated.
2. Branch displacement mutation (BDM), where a number of branches is displaced along the parent branch. BDM is controlled by two parameters – number of branches to be displaced (d_B) and the range of displacement (d_R).

In general, the mutation rate should be relatively low. The fine tuning is usually done by experimentation.

2.5. Fitness (cost) function

Formulating the fitness function, which in minimization problems, as in this case, is called cost function (CF) is usually the most challenging part of the EA implementation. It does not only require knowledge particular to the considered problem, but it is also necessary to give univocal, usually numerical, evaluation of each candidate solution, including the unacceptable ones. CF in the given environment E for the sequence of TZ units *S* is formulated as an aggregate objective function (AOF), that is as weighted linear combination of parameters, as follows:

$$CF_S^E = Minimize\left(w_1 n_u + w_2 \frac{\sum_{t=1}^{T} Min[d_t]}{n_{UE}} + w_3 \sum_{v=1}^{V} P_v\right)$$

where, n_u is the number of units in the TZ network,
d_t is the distance from an i^{th} unit of the TZ network to the t^{th} terminal,
T is the number of terminals, since the network starts from the terminal 1 – it is omitted in the calculations.
n_{UE} is the number of unique ends towards terminals of the TZ network,
P_v is the penalty of the vertex v violating an obstacle, V is the number of vertices violating the obstacles.
w_1, w_2, w_3 are the parametric weights.

The penalty is applied to the configurations where vertices of some trapezoidal units violate the obstacle areas, as visualized in figure 5.

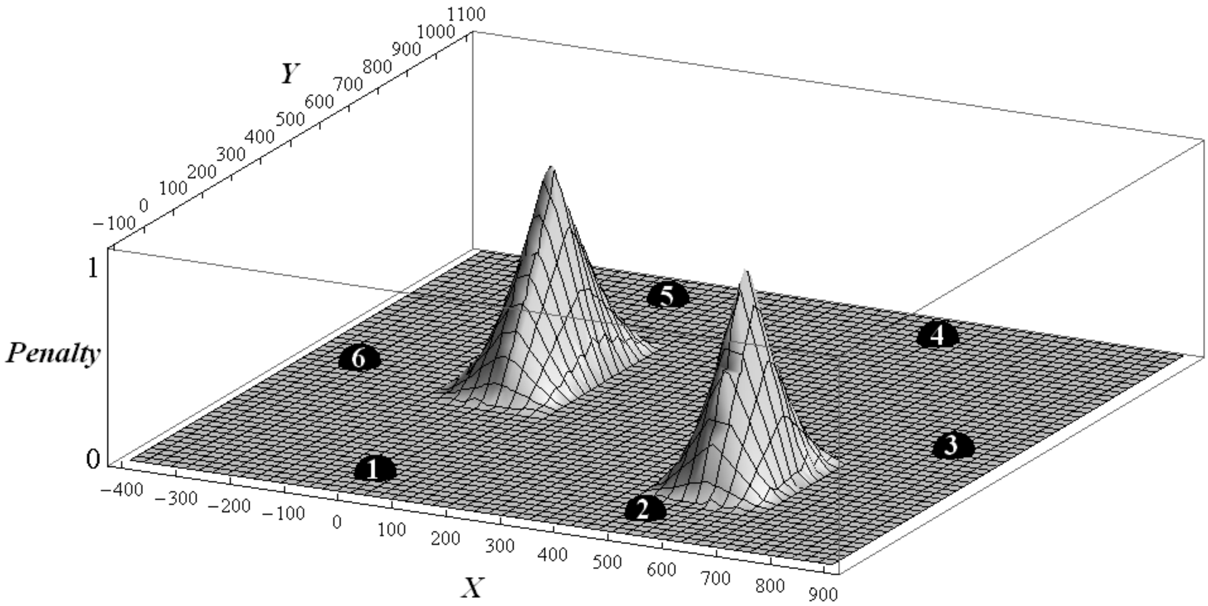
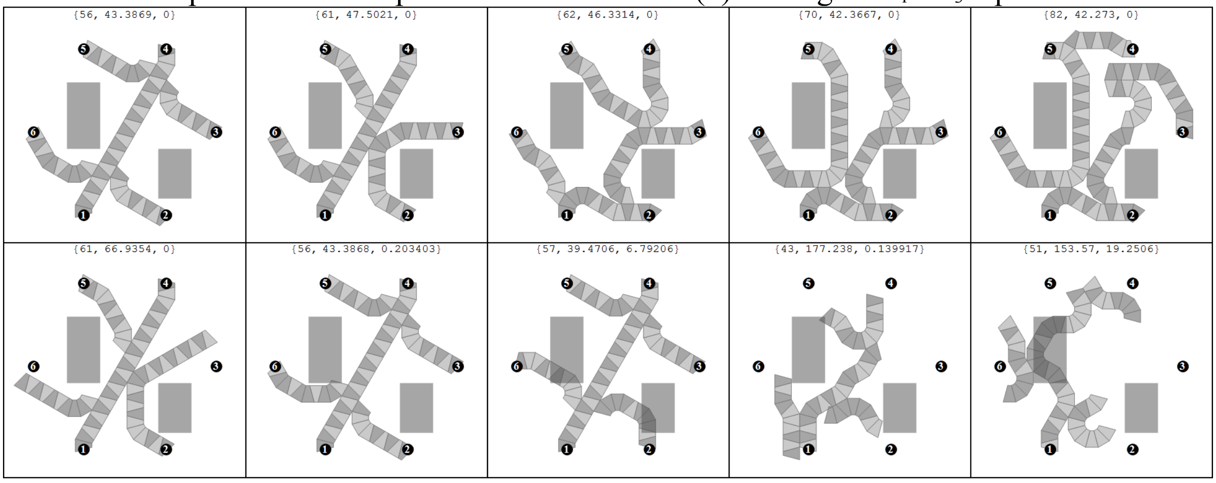


Figure 5. The penalty is the function of the coordinates of the units' vertices. If a vertex lies inside an obstacle, the closer it is located to the obstacle's centroid - the higher is the penalty, otherwise there is no penalty. The positions of terminals are indicated by black dots.

In order to calibrate the weights $w_1 ... w_3$, a set of representative solutions was prepared as shown in table 2.

Table 2. Sample solutions from the best to the worst. The three values given for each one correspond to the components of the formula (1) at weights $w_1 \ldots w_3$ equal to 1.

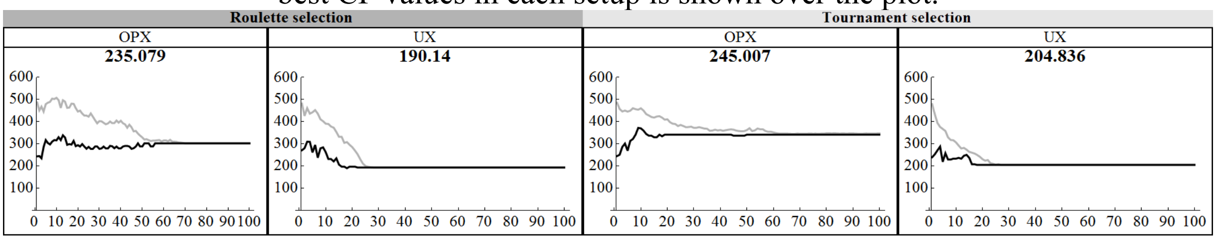


The weights $w_1 \ldots w_3$ from the formula (1) where adjusted, so the values of CF monotonically increase with the decreasing quality of a solution. The values are: $w_1 = 0.24$, $w_2 = 0.72$, $w_3 = 10$.

2.6. The experiments

In order to set up the parameters for EA, initially a number of small experiments have been performed. The first experiment was done for two types of selection and two types of crossover (no mutation). The results are shown in table 3.

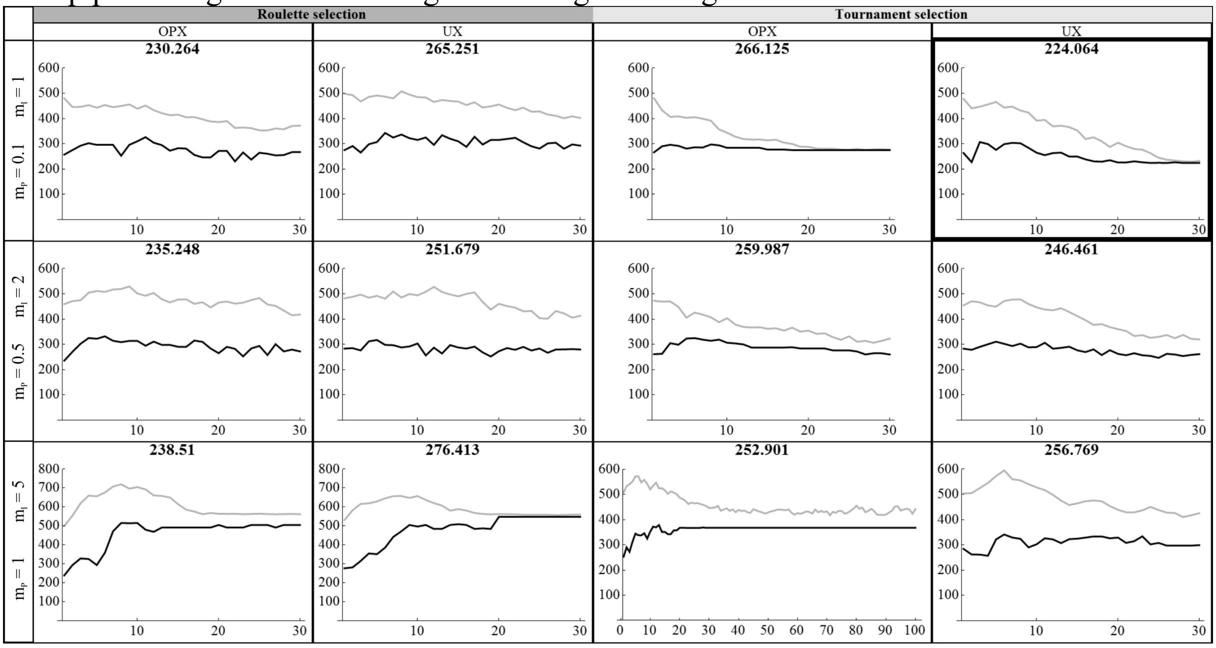
Table 3. The average and the minimal CF of the population are shown in gray and black respectively. The size of population is 100 and so is the number of generations. The overall best CF values in each setup is shown over the plot.



As table 3 indicates, in this setup the uniform crossover (UX) gives better results than the one-point crossover (OPX). Moreover, as expected, that the lack of mutation leads to degeneration of the population. After a certain, usually rather small number of generations, there are no improvements due to the homogeneity of the candidate solutions. Table 4 collects the results of the experiments with introduced unit switch mutation (USM).

Table 4. Series of experiments with USM and various combinations of EA parameters. The setup producing the best convergence throughout the generations is framed with a thick line.

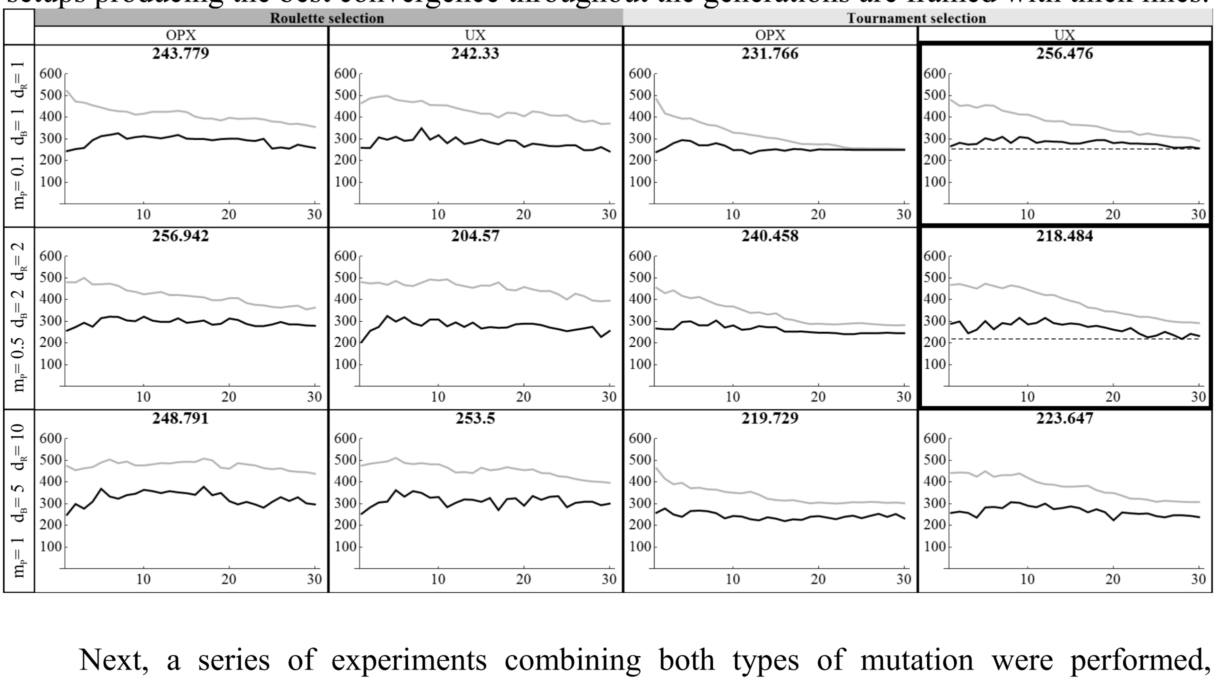
	Roulette selection		Tournament selection	
	OPX	UX	OPX	UX
$m_P = 0.1$ $m_I = 1$	230.264	265.251	266.125	224.064
$m_P = 0.5$ $m_I = 2$	235.248	251.679	259.987	246.461
$m_P = 1$ $m_I = 5$	238.51	276.413	252.901	256.769



Next, the branch displacement mutation (BDM) only was implemented. Table 5 collects the results of the experiments.

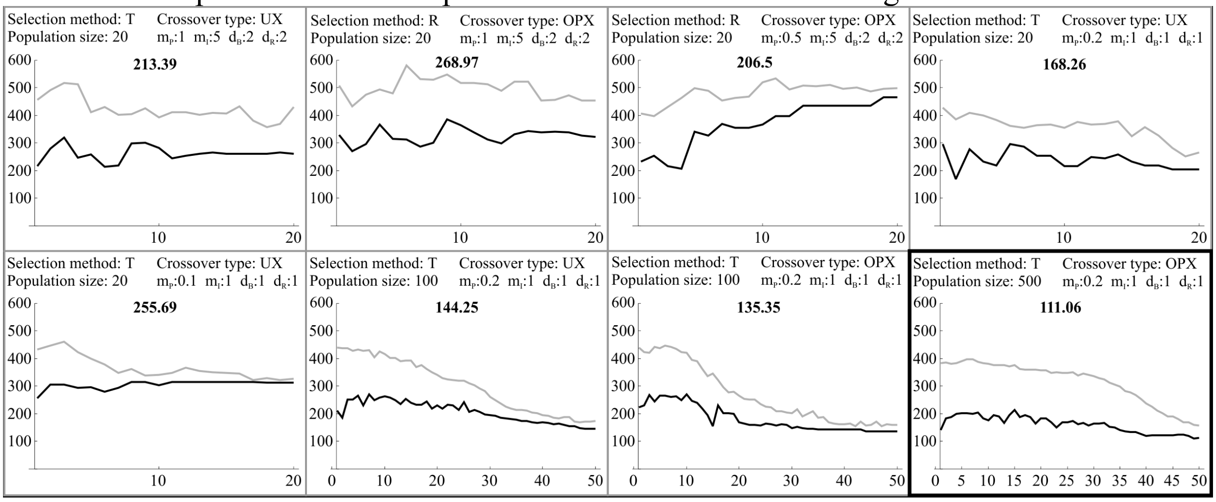
Table 5. Series of experiments with BDM and various combinations of EA parameters. The setups producing the best convergence throughout the generations are framed with thick lines.

	Roulette selection		Tournament selection	
	OPX	UX	OPX	UX
$m_P = 0.1$ $d_B = 1$ $d_R = 1$	243.779	242.33	231.766	256.476
$m_P = 0.5$ $d_B = 2$ $d_R = 2$	256.942	204.57	240.458	218.484
$m_P = 1$ $d_B = 5$ $d_R = 10$	248.791	253.5	219.729	223.647



Next, a series of experiments combining both types of mutation were performed, followed by the final experiment. The results are shown in table 6.

Table 6. A series of experiments with both USM and BDM at different EA parameters. The plot for the final experiment is shown in the bottom right corner.



The result of the final experiment is shown and compared with the solution created manually and generated by backtracking in figure 6.

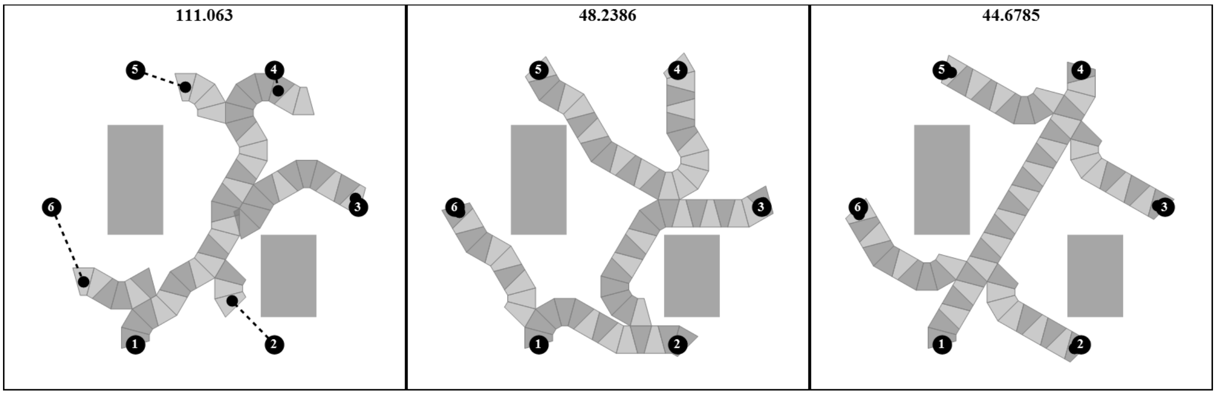


Figure 6. From the left: The TZ network produced by EA, backtracking and manually. The CF is shown above each solution.

3. CONCLUSIONS

Presently, the results of EA to this complex optimization problem are not fully satisfactory. However, this project is not yet completed and significant improvements are expected in the near future. Besides executing more experiments for fine-tuning of the EA parameters, the CF is to be improved to reflect the quality of solutions more adequately. Moreover, as it was already implemented for a single branch in [5] the units that “go away” from the closest proximity of the end terminal to be ignored for the CF calculations. Also the encoding of the genotypes into a one dimensional list is under consideration – this would make the genetic operations more straightforward.

Acknowledgments

This is part of a postdoctoral project grant-funded by the Japanese Society for the Promotion of Science. The research is titled "Improvements of the Seniors' Quality of Life through Application of Innovative Computational Systems to the problems of Accessibility, Ergonomics and Housing & Living Environment".

4. REFERENCES

[1] Zawidzki M., Nishinari K., Modular Truss-Z system for self-supporting skeletal free-form pedestrian networks, Advances in Engineering Software, vol. 47, no.1, pp. 147–159, 2012.

[2] Zawidzki M. A Modular Truss System for Pedestrian Traffic and a Computer Implementation of an Algorithm for Creating the Truss Structure, in B.H.V. Topping, J.M. Adam, F.J. Pallarés, R. Bru, M.L. Romero, (Editors), Proceedings of the Tenth International Conference on Computational Structures Technology, Civil-Comp Press, Stirlingshire, UK, doi:10.4203/ccp.93.363, 2010.

[3] Zawidzki M. Nishikawa I. Discrete optimization of modular truss network in constrained environment, Proceedings for CJK-OSM6: 6th China-Japan-Korea Joint Symposium on Optimization of Structural and Mechanical Systems, Kyoto, Japan, 2010.

[4] Zawidzki M. Tiling a path with a single trapezoid along a given curve – an interactive demonstration where a trapezoidal tiling – an equivalent to the projection of the run of the truss on the XY plane is dynamically aligned. http://demonstrations.wolfram.com/TilingAPathWithASingleTrapezoidAlongAGivenCurve/

[5] Zawidzki M. Tateyama K. Application of Evolution Strategy for Minimization of the Number of Modules in a Truss Branch Created with the Truss-Z System, in Tsompanakis Y, Topping BHV. (Eds), Proceedings of the Second International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering, Civil-Comp Press, Stirlingshire, UK, Paper 9, doi:10.4203/ccp.97.9, 2011.