

Autoreferat

1. Imię i Nazwisko

Maciej (Machi)¹ Zawidzki
Instytut Podstawowych Problemów Techniki PAN
Zakład Technologii Inteligentnych
ul. Pawińskiego 5B, 02-106 Warszawa
E: zawidzki@MIT.edu

2. Posiadane dyplomy, stopnie naukowe z podaniem nazwy, miejsca i roku ich uzyskania oraz tytułu rozprawy doktorskiej

Doctor of Engineering – Ritsumeikan University, Kioto, Japonia 2010. Tytuł rozprawy:
Application of Computational Intelligence to engineering design problems in Architecture-firmitatis, utilitatis, venustatis.

Nostryfikacja stopnia doktorskiego przez Instytut Podstawowych Problemów Techniki PAN w dyscyplinie **Informatyka** 28 stycznia 2016.

3. Informacje o dotychczasowym zatrudnieniu w jednostkach naukowych

- Od 2/2019 Adiunkt w Zakładzie Technologii Inteligentnych w Instytucie Podstawowych Problemów Techniki PAN (umowa na czas nieokreślony).
- 2/2017 – 1/2019 Adiunkt w ZTI IPPT PAN – kierownik projektu pt. *Innovative Extremely Modular Systems for temporary and permanent deployable structures and habitats: development, modeling, evaluation & optimization*, w ramach dwuletniego grantu badawczego “**Polonez 2**” finansowanego przez Narodowe Centrum Nauki. Umowa nr. 2016/21/P/ST8/03856.
- 10/2016 – 2/2017 Adiunkt na Wydziale Inżynierii Produkcji w Szkole Głównej Gospodarstwa Wiejskiego w Warszawie.
- 12/2014 – 11/2015 Kierownik w autorskim dwuletnim programie badawczym pt. *Effective computational methods for grid & raster-based modeling of practical problems in architectural & urban layout design* w ramach stażu podoktorskiego *Singapore University of Technology and Design – Massachusetts Institute of Technology Postdoctoral Program* w Singapurze.
- 12/2013 – 11/2014 Kierownik w w/w programie badawczym w School of Architecture of the **Massachusetts Institute of Technology**, Cambridge, USA.

¹ Ponieważ zarówno doktorat jak i pierwszy staż podoktorski odbyłem w Japonii, prace sygnowałem fonetyczną transkrypcją imienia Maciej czyli Machi, która była zrozumiała w moim ówczesnym środowisku. By nie komplikować później sytuacji pozostałem konsekwentnie przy tej konwencji.

10/2011 – 09/2013 Kierownik w autorskim dwuletnim programie badawczym pt. *Improvements of the Seniors' Quality of Life through Application of Innovative Computational Systems to the problems of Accessibility, Ergonomics and Housing & Living Environment* finansowanym przez **Japan Society for Promotion of Science** w Department of Mathematical Physics of Emergent Systems, Research Center for Advanced Science and Technology (**RCAST**) – **University of Tokyo** w Japonii

09/2010 – 04/2011 Asystent w Ritsumeikan University w Japonii. Tytuł badań autorskich: *Development and Simulation of Computational Optimization Methods in Architecture*.

4. Wskazanie osiągnięcia wynikającego z art. 16 ust. 2 ustawy z dnia 14 marca 2003 r. o stopniach naukowych i tytule naukowym oraz o stopniach i tytule w zakresie sztuki (Dz. U.2017 r. poz. 1789):

a) tytuł osiągnięcia naukowego

Zastosowania metod inteligencji obliczeniowej do optymalizacji architektonicznych Systemów Ekstremalnie Modularnych

b) (autor/autorzy, tytuł/tytuły publikacji, rok wydania, nazwa wydawnictwa, recenzenci wydawniczy)

1. M. Zawidzki, 2017, [*Discrete Optimization in Architecture – Extremely Modular Systems*](#), SpringerBriefs in Architectural Design and Technology, Springer Singapore. Monografia, rozdziały indeksowane przez **SCOPUS**.
2. M. Zawidzki, K. Nishinari, 2012, [*Modular Truss-Z system for self-supporting skeletal free-form pedestrian networks*](#), **ADVANCES IN ENGINEERING SOFTWARE** 47(1), pp.:147-159. **IF₅:2.54, Q1** (Computer Science, Interdisciplinary Applications) **Q1** (Computer Science, Software Engineering) **Q1** (Engineering, Multidisciplinary), **Lista A MNiSW – 30 pkt (dblp)**.
3. M. Zawidzki, K. Nishinari, 2013, [*Application of evolutionary algorithms for optimum layout of Truss-Z linkage in an environment with obstacles*](#), **ADVANCES IN ENGINEERING SOFTWARE** 65 pp.:43-59. **IF₅:2.54, Q1** (Computer Science, Interdisciplinary Applications) **Q1** (Computer Science, Software Engineering) **Q1** (Engineering, Multidisciplinary), **Lista A MNiSW – 30 pkt (dblp)**.
4. M. Zawidzki, 2015, [*Retrofitting of pedestrian overpass by Truss-Z modular systems using graph-theory approach*](#), **ADVANCES IN ENGINEERING SOFTWARE** 81 pp.:41-49. **IF₅:2.54, Q1** (Computer Science, Interdisciplinary Applications) **Q1** (Computer Science, Software Engineering) **Q1** (Engineering, Multidisciplinary), **Lista A MNiSW – 30 pkt (dblp)**.

5. M. Zawidzki, 2016, [*Optimization of Multi-branch Truss-Z based on evolution strategy*](#), **ADVANCES IN ENGINEERING SOFTWARE 100** pp.:113-125. **IF5:2.54, Q1** (Computer Science, Interdisciplinary Applications) **Q1** (Computer Science, Software Engineering) **Q1** (Engineering, Multidisciplinary), **Lista A MNiSW – 30 pkt. (dblp)**
6. M. Zawidzki, J. Szklarski, 2018, [*Effective multi-objective discrete optimization of Truss-Z layouts using a GPU*](#), **APPLIED SOFT COMPUTING 70** pp.:501-512. **IF5:4.004, Q1** (Computer Science, Artificial Intelligence) **Q1** (Computer Science, Interdisciplinary Applications), **Lista A MNiSW – 40 pkt. (dblp)**
7. M. Zawidzki, J. Szklarski, 2018, [*Transformations of Arm-Z modular manipulator with Particle Swarm Optimization*](#), **ADVANCES IN ENGINEERING SOFTWARE 126**, pp.:147-160. **IF5:2.54, Q1** (Computer Science, Interdisciplinary Applications) **Q1** (Computer Science, Software Engineering) **Q1** (Engineering, Multidisciplinary), **Lista A MNiSW – 30 pkt. (dblp)**
8. M. Zawidzki, Ł. Jankowski, 2018, [*Optimization of modular Truss-Z by minimum-mass design under equivalent stress constraint*](#), **SMART STRUCTURES & SYSTEMS, 21(6)**, pp.715-725. **IF5: 1.43), Q1** (Engineering, Civil) **Q2** (Engineering, Mechanical) **Q2** (Instruments & Instrumentation), **Lista A MNiSW – 30 pkt.**

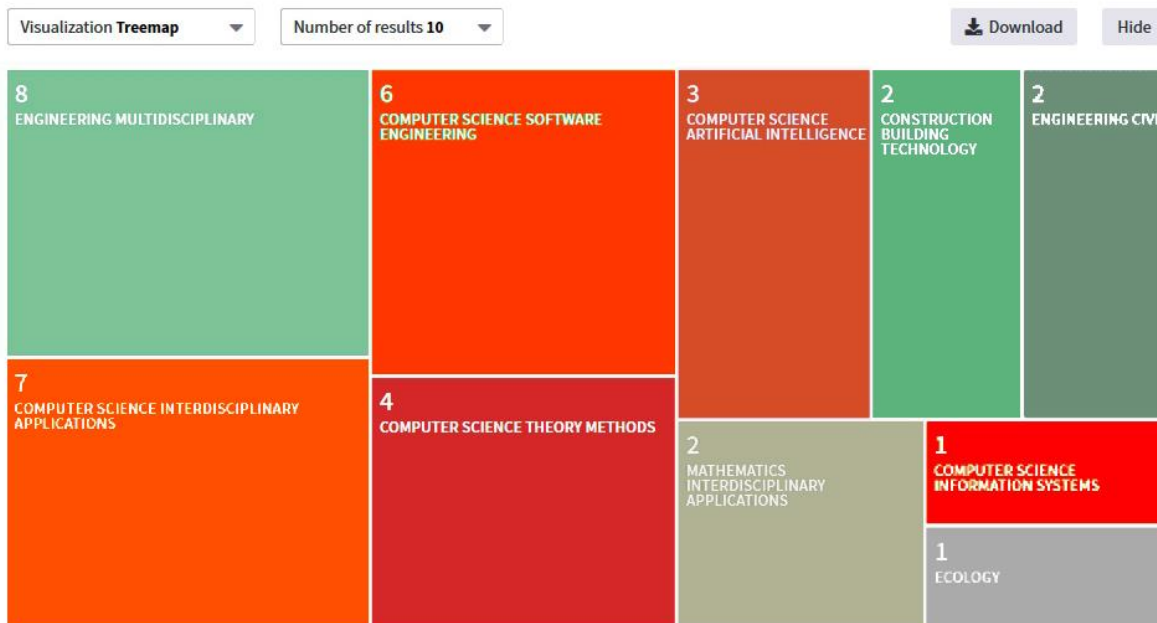
c) **omówienie celu naukowego ww. prac i osiągniętych wyników wraz z omówieniem ich ewentualnego wykorzystania**

Habilitant jest z zawodu architektem. Niemniej po kilku latach pracy zawodowej „skręcił” w stronę nauki – najpierw „inżynierskiej”, potem stopniowo w stronę informatyki. Nie była to bynajmniej ucieczka przed architekturą, lecz głęboka potrzeba stworzenia czegoś pożytecznego, fundamentalnie wartościowego – czy to w sensie praktycznym czy poznawczym.

W doktoracie pt. **Zastosowanie metod inteligencji obliczeniowej do problemów projektowania inżynierskiego w Architekturze – *utilitatis*** (łac. użyteczność), ***firmitatis*** (łac. trwałość), ***venustatis*** (łac. piękno)², zgodnie z tytułem, habilitant wskazał trzy konkretne obszary zastosowania metod inteligencji obliczeniowej w architekturze i naszkicował pewne twórcze rozwiązania. Rozwiązania te miały dwie zasadnicze zalety: innowacyjność (czego dowodzą publikacje naukowe) i obiektywizm, czyli możliwość liczbowej jednoznacznej oceny ich jakości wyrażonej funkcją celu. Przyporządkowanie wielu współczesnych badań do konkretnych i wyłącznych dyscyplin jest często problematyczne. Tym trudniejsze w przypadku tak interdyscyplinarnych badań jak te prezentowane tutaj. Świadczyć o tym choćby może fakt, że habilitant zdobył w Japonii tytuł *Doctor of Engineering*, zaś dyplom został w Polsce uznany za równoważny z polskim dyplomem doktora nauk technicznych w dyscyplinie Informatyka. I jest to rozpoznanie jak najbardziej aktualne i pokrywa się z charakterem publikacji habilitanta co ilustruje poniższy rysunek.

² (oryg.: *Application of Computational Intelligence to engineering design problems in Architecture- firmitatis, utilitatis, venustatis*)

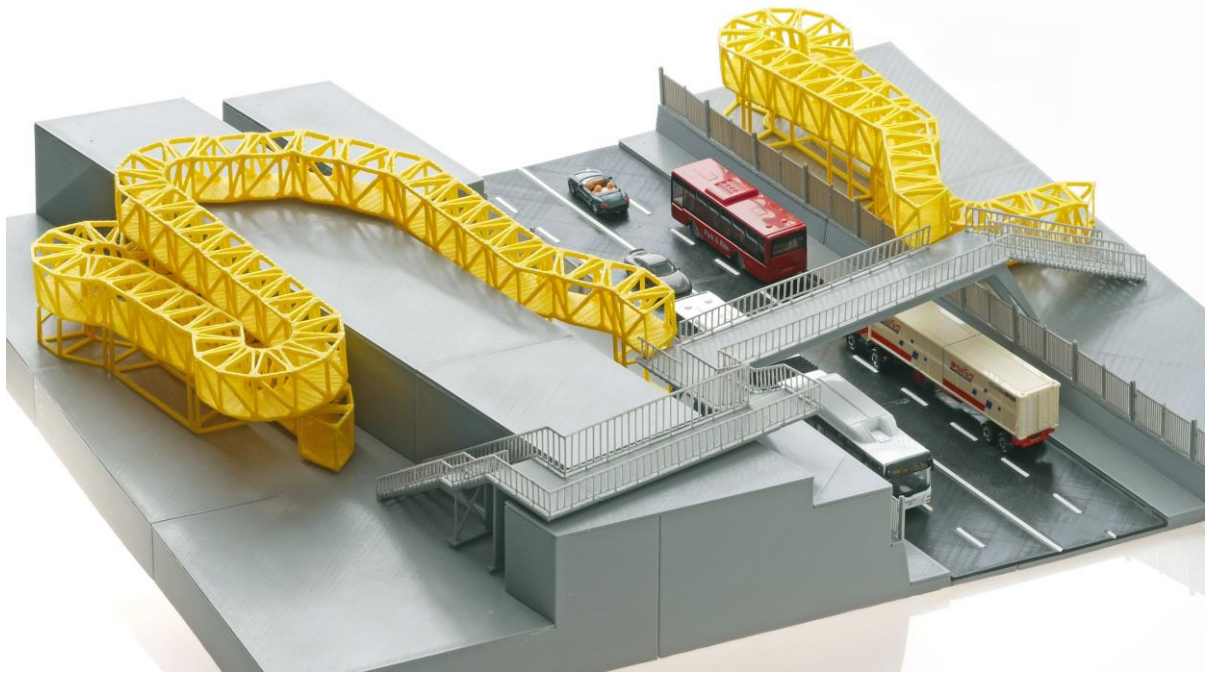
Showing 19 records for AUTHOR: (zawidzki m. NOT Walega NOT Schmidt)

[Create Citation Report](#)

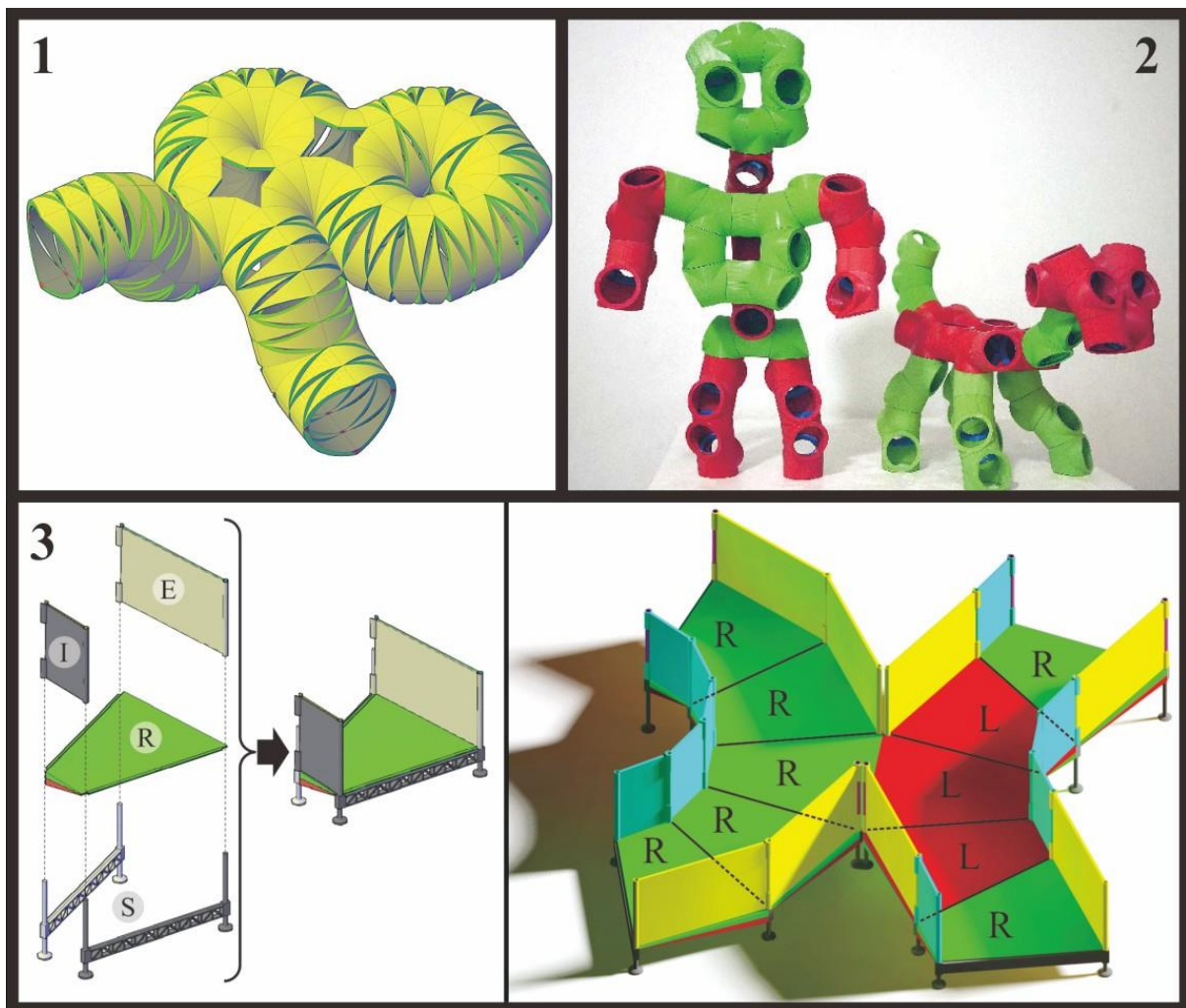
Badania naukowe habilitanta po doktoracie w dyscyplinie informatyka koncentrują się na aplikacjach odnoszących się do problemów projektowania architektonicznego i urbanistycznego, przeto mają charakter wielowątkowy. Niemniej daje się w nich wyodrębnić obszar, będący spójnym osiągnięciem naukowym, który jest treścią niniejszego wniosku.

Jednym z trzech nowatorskich i kreatywnych rozwiązań przedstawionych w rozprawie doktorskiej był system *Truss-Z* (w skrócie TZ). Czyli system kratownicowo-ramowy, składający się z jednego rodzaju modułu (i jego lustrzanego odbicia) umożliwiający tworzenie pochylni o stałym koncie nachylenia i dowolnym kształcie, co ilustruje Fig.1.

Po obronie doktoratu habilitant opracował szereg narzędzi informatycznych do systemu TZ oraz kilku stworzonych później systemów składających się z jednego typu modułu i umożliwiających konstruowanie skomplikowanych struktur. Systemy te to chronologicznie: Pipe-Z (Fig.21), jego późniejsze rozwinięcie - Arm-Z (Fig.31), „składalne” (ang. foldable) Pipe-Z (Fig.32), wielogałęziowe (ang. multibranch) Truss-Z (Fig.14), wielogałęziowe Pipe-Z (Fig. 2.2), Ramp-Z (Fig.2.3) i Vault-Z (Fig.2.1).



1. Fotografia modelu fizycznego w skali 1:100 rozwiązania pochylni modularnej *Truss-Z*. Konfiguracja została wygenerowana przy pomocy algorytmu grafowego (wyjaśnienie dalej w tekście).



2. 1) Vault-Z. 2) Multibranch Pipe-Z. 3) Ramp-Z.

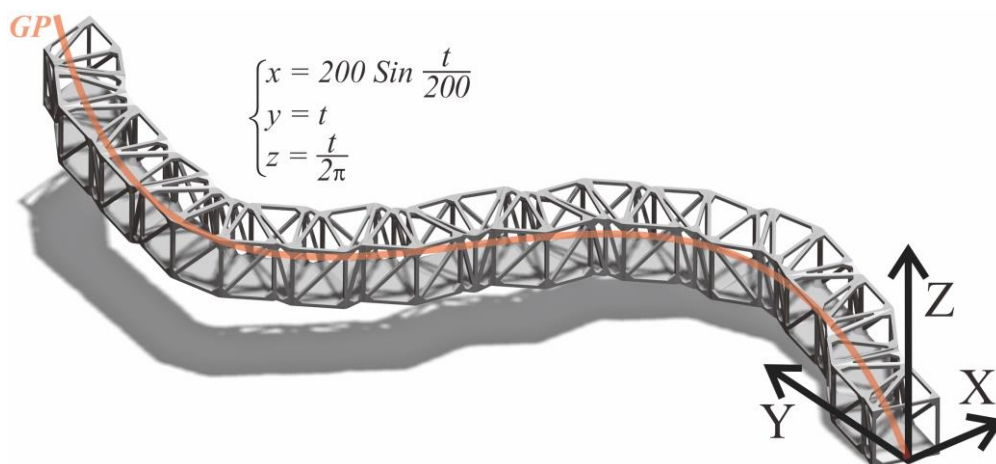
Z czasem na określenie tego typu koncepcji konstrukcyjnych zaproponowano termin „System Ekstremalnie Modułarny” (ang. Extremely Modular System, w skrócie EMS).

EMSy mają cztery zasadnicze zalety:

- ekonomiczną – ponieważ dają możliwość masowej prefabrykacji (przez co mogą być niedrogie i szeroko stosowane);
- funkcjonalną – ponieważ dają możliwość rekonfiguracji, rozbudowy, redukcji, szybkiego montażu;
- niezawodność – łatwość ewentualnej naprawy, gdyż każdy zepsuty moduł może być zastąpiony identycznym;
- naukową – ponieważ mogą być inteligentnie modelowane matematycznie.

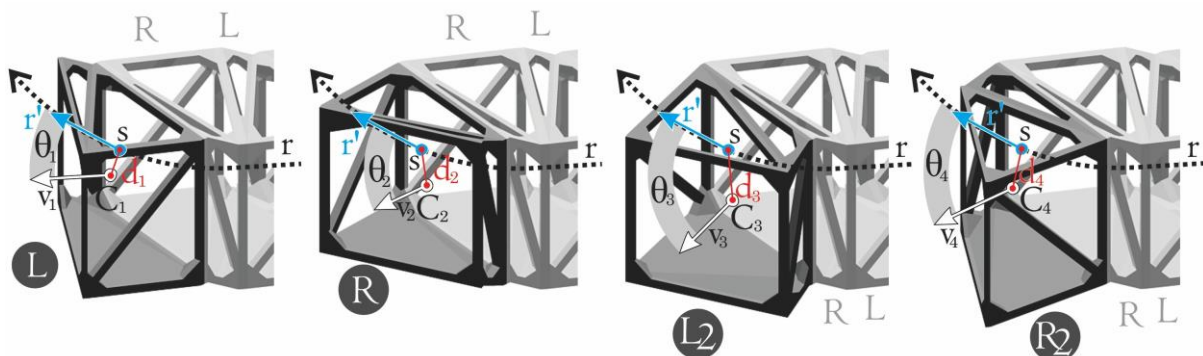
Systemy te mają wszak jedną zasadniczą wadę – są nieintuicyjne, innymi słowy są trudne do manualnego (bezpośredniego) konfigurowania. Liczba wszystkich możliwych kombinacji modułów danego typu szybko „eksploduje” wraz z ich rosnącą liczbą i dla realistycznych przypadków, wybór najlepszego rozwiązania spośród wszystkich jest po prostu nierealny. Aby móc wykorzystywać EMSy w sposób celowy i racjonalny, z pomocą przychodzą metody optymalizacyjne a do ich efektywnej realizacji – metody informatyczne. Temu zagadnieniu habilitant poświęcił monografię pt. „Optymalizacja Dyskretna w Architekturze – Systemy Ekstremalnie Modułarne” (ang. „Discrete Optimization in Architecture – Extremely Modular Systems”) [1].

Dwa lata po uzyskaniu stopnia doktora, w roku 2012, opublikowany zostaje **pierwszy artykuł niniejszego cyklu habilitacyjnego** [2]. We wstępie przeprowadza dyskusję na temat konfliktu między modularnością a dowolnością formy architektonicznej. Wynika z niej zasadniczo, że dostępne systemy konstrukcyjne albo są wysoce modułarne i oferują nieciekawe, repetytywne formy, albo oferują ciekawe formy, lecz wymagają niestandardowych rozwiązań (ang. customization). W tym kontekście wprowadzony zostaje system Truss-Z, który z jednej strony jest *ekstremalnie* modułarny, a z drugiej oferuje dowolność kształtowania formy. Artykuł dalej wprowadza formalny opis modułu TZ, i poddaje go wstępnej analizie statycznej, przy pomocy technik grafowych opisuje jego cechy topologiczne, określa stopień statycznej niewyznaczalności i wykazuje w oparciu o lemat Cauchy’ego jego geometryczną niezmienniczość. Zostają przedstawione dwa algorytmy do automatycznego tworzenia jednogalęziowych konstrukcji TZ: poprzez dopasowanie całej struktury do zadanej ścieżki w przestrzeni oraz poprzez implementację algorytmu z nawrotami (ang. *backtracking*). Pierwszy wymaga znajomości ścieżki (ang. guide path, *GP*, patrz Fig.3) wzdłuż której ma podążać TZ i sformułowany jest jako problem minimalizacji.



3. Przykładowa ścieżka GP podana parametrycznie i 33 moduły TZ optymalnie podążające za nią.

Moduł TZ można dołączyć na 4 sposoby (Fig. 4).

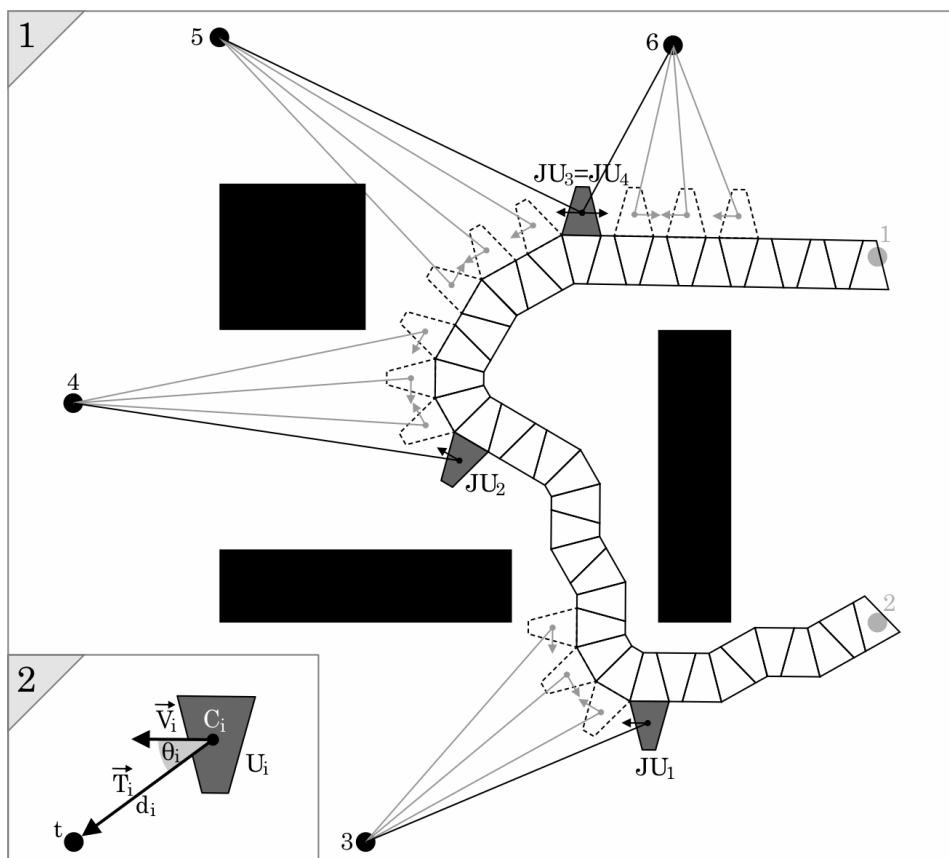


4. Cztery sposoby dołączania modułu TZ: R, L (odbicie lustrzane R), L2 (obrót odbicia L) i R2 (obrót R).

Algorytm wybiera taką sekwencję modułów by suma ważona błędów: odległości d (centroidy każdego modułu od GP) i kątów θ (między kierunkami modułów i GP) była najmniejsza (1).

$$\text{Minimize} \left(\frac{a}{b} d_i + (1-a)(1 - v_i \cdot r'[s]) \right) \quad (1)$$

gdzie, d_i to najmniejsza odległość centroidy (C_i) i -tego modułu od punktu s na krzywej r (GP), v_i to wektor i -tego modułu, $r'[s]$ to kierunek GP r w punkcie s , a i b to parametry (wagi). Zaimplementowano tę procedurę w rozbudowanej formie również do struktury wielogłęziowej. Fig. 5 ilustruje sposób wyznaczania punktów rozgałęzień (ang. junction unit, JU).

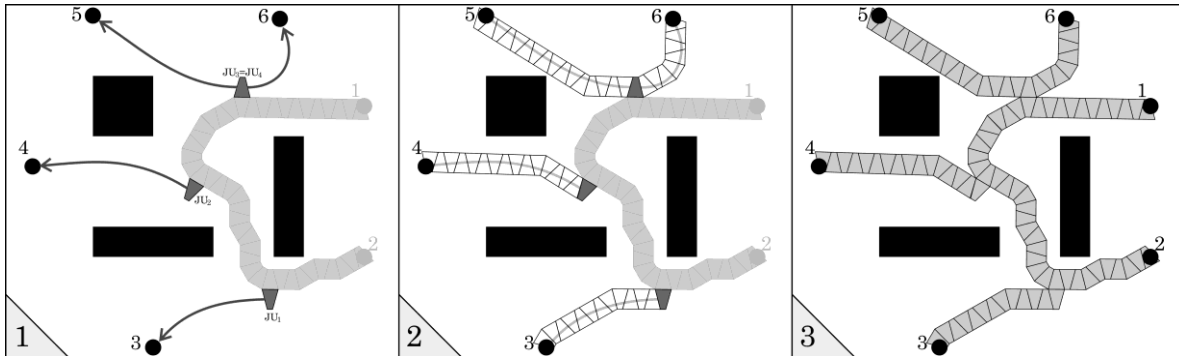


5. 1) Potencjalne punkty rozgałęzień zaznaczono linią przerywaną. 2) Wielkości minimalizowane przy wyborze punktu rozgałęzienia.

Zmodyfikowana funkcja (1) do lokalizowania punktów rozgałęzień:

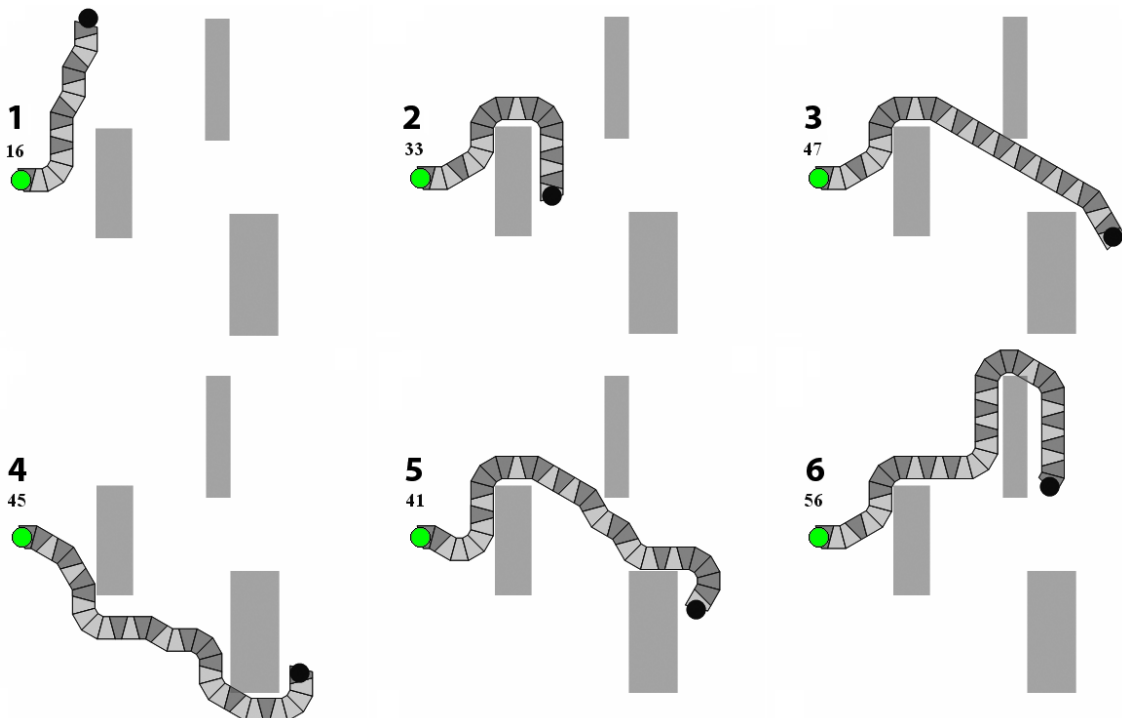
$$\text{Minimize} \left(\frac{a}{b} d_i + (1-a) \cdot (\vec{T}_i \cdot \vec{V}_i) \right) \quad (2)$$

gdzie, d_i to odległość między centroidą (C_i) i-tego potencjalnego modułu rozgałęzienia i terminalem t , \vec{V}_i to wektor tego modułu, \vec{T}_i to kierunek z centroidy (C_i) do terminalu t , a i b to parametry jak w (1). Fig. 6 ilustruje pozostałe kroki algorytmu.



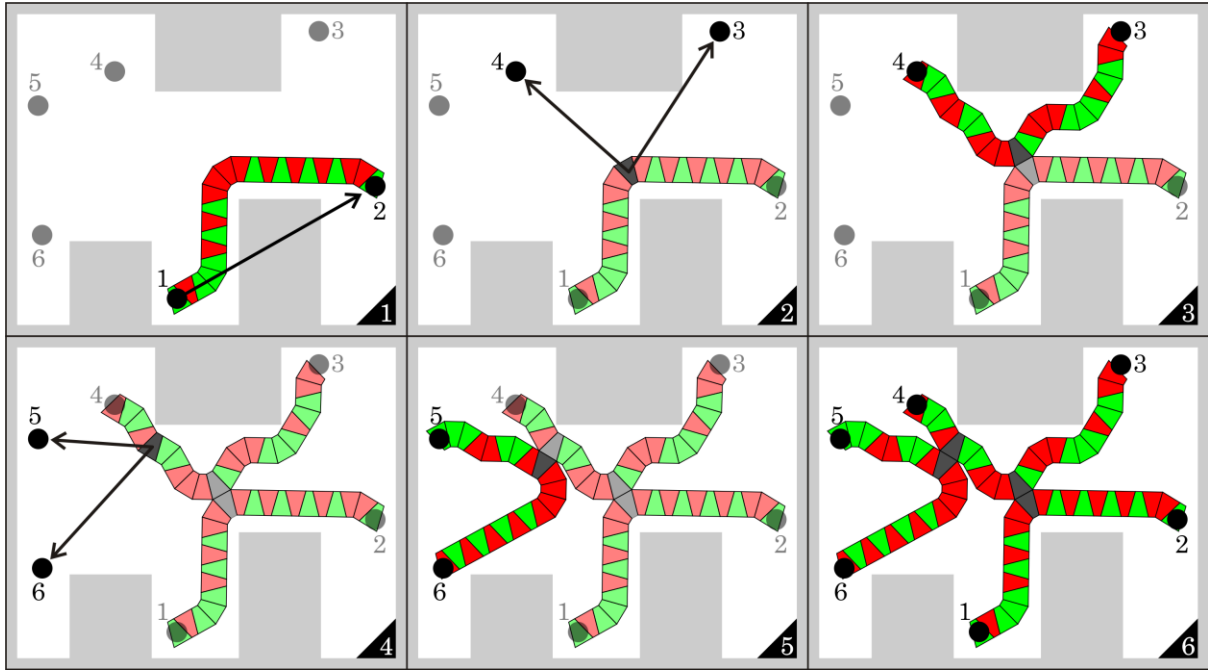
6. Trzy schematyczne kroki w budowie wielogłęziowej struktury TZ przy pomocy ścieżek GP.

Drugi przedstawiony algorytm uwzględnia zakaz kolizji z przeszkodami (choć nie uwzględnia auto-kolizji). Kolizje są niedopuszczalne (ang. „death penalty”). Procedura oparta jest na algorytmie z nawrotami i zawsze generuje rozwiązania dopuszczalne, choć rzadko idealne (Fig. 7).



7. Widok z góry przykładowych struktur TZ wygenerowanych przez *backtracking*.

Zaimplementowano tę procedurę w rozbudowanej formie również do struktury wielogłęziowej co ilustruje Fig. 8



8. Procedura tworzenia sieci TZ łączącej 6 terminali: 1) połączenie pierwszych dwóch terminali, 2) wyszukanie dwóch kolejnych terminali i stworzenie punktu rozgałęzienia, 3) wypuszczenie dwóch gałęzi TZ przy pomocy *backtrackingu*, 4) jak w 2), 5) jak w 3), 6) kompletna sieć TZ.

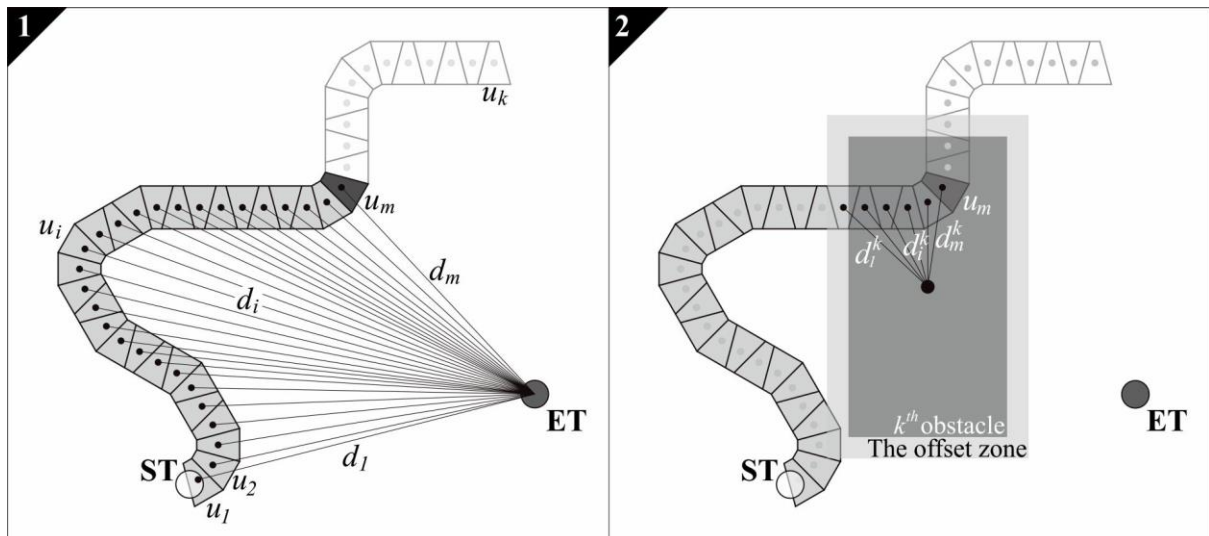
Drugi artykuł w tym cyklu [3] wprowadza nomenklaturę oraz kodowanie konstrukcji TZ w genotyp w celu umożliwienia zastosowania metaheurystyk, oraz metodę jego kompresji w oparciu o system 36-owy. Sformułowana jest wielokryterialna funkcja celu (kosztu) CF_s w której skład wchodzi: minimalizacja liczby modułów TZ, minimalizacja odległości ostatniego modułu od zadanego terminalu i minimalizacja kolizji konstrukcji TZ z elementami otoczenia.

$$CF_s = G_s \times P_s \quad (3)$$

$$G_s = \frac{w_1 \sum_{i=1}^m d_i}{m} + w_2 \min\{d_1 \dots d_m\} + w_3 m \quad (4)$$

$$P_s = 1 + \frac{w_4 \sum_{k=1}^U c_k}{\sum_{k=1}^U A_k} \sum_{k=1}^U \frac{A_k}{1 + \log_{\frac{3}{2}}[1 + \min\{d_1^k \dots d_i^k \dots d_m^k\}]} \quad (6)$$

Fig. 9 ilustruje obliczanie składowych G_s i P_s

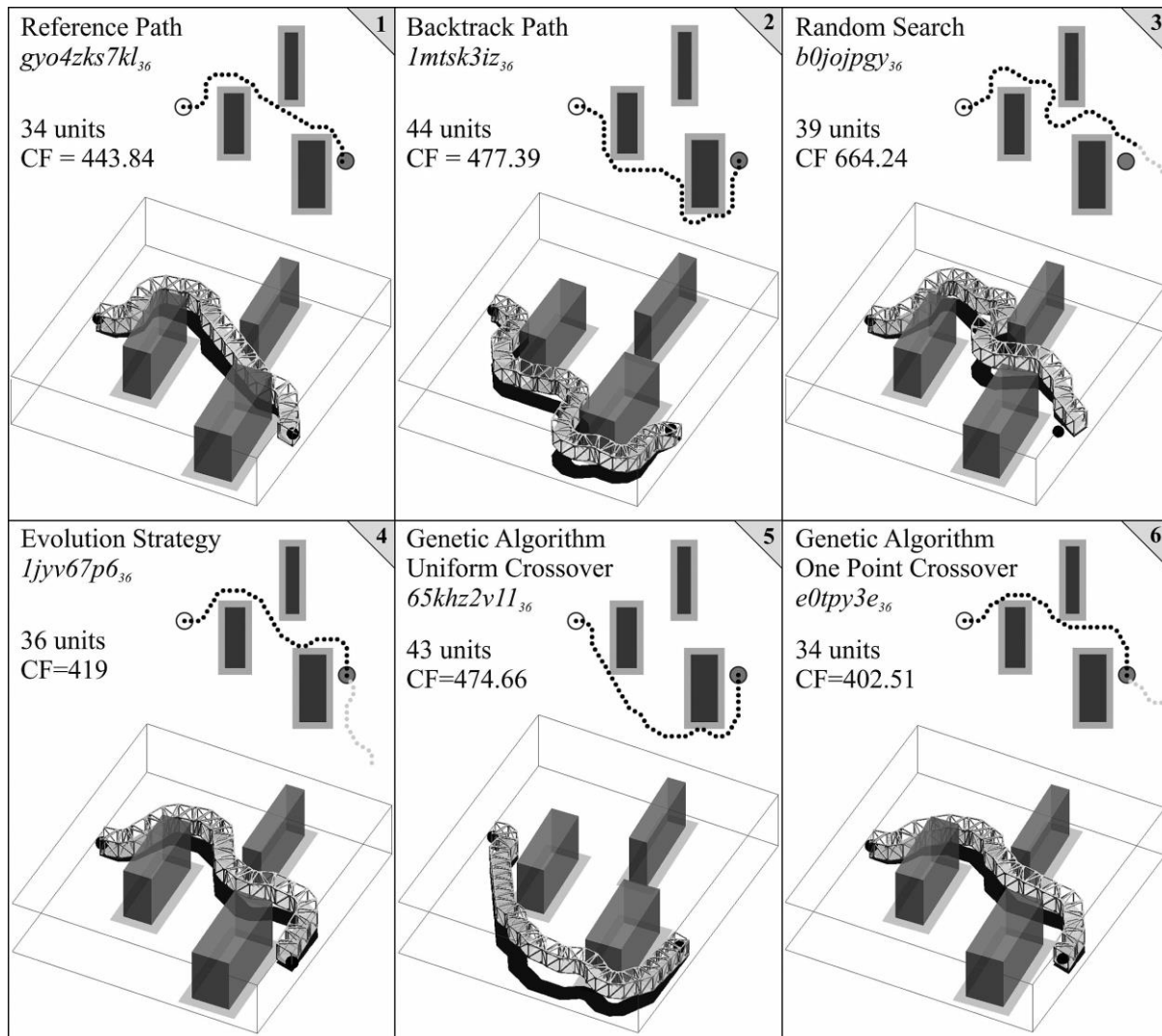


9. 1) G_s „nagradza” za przybliżanie się do celu (ET). 2) P_s „karze” za kolizje z przeszkodami – im więcej kolizji, tym wyższa kara.

Opracowano operacje mutacji i krzyżowania, i zaimplementowano klasyczne metaheurystyki: *strategię ewolucyjną*³ (czyli względnie silna mutacja przy braku krzyżowania) oraz *algorytm genetyczny* (krzyżowanie najlepszych kandydatów przy względnie małej mutacji) w kilku wariantach parametrów w celu znalezienia najlepszego rozwiązania dla referencyjnego układu początkowego (ST) i końcowego (ET) terminalu i 3 przeszkód. Skuteczność i ogólność metody wykazano na dodatkowych dwóch przykładach. Wyniki ilustruje Tab.1.

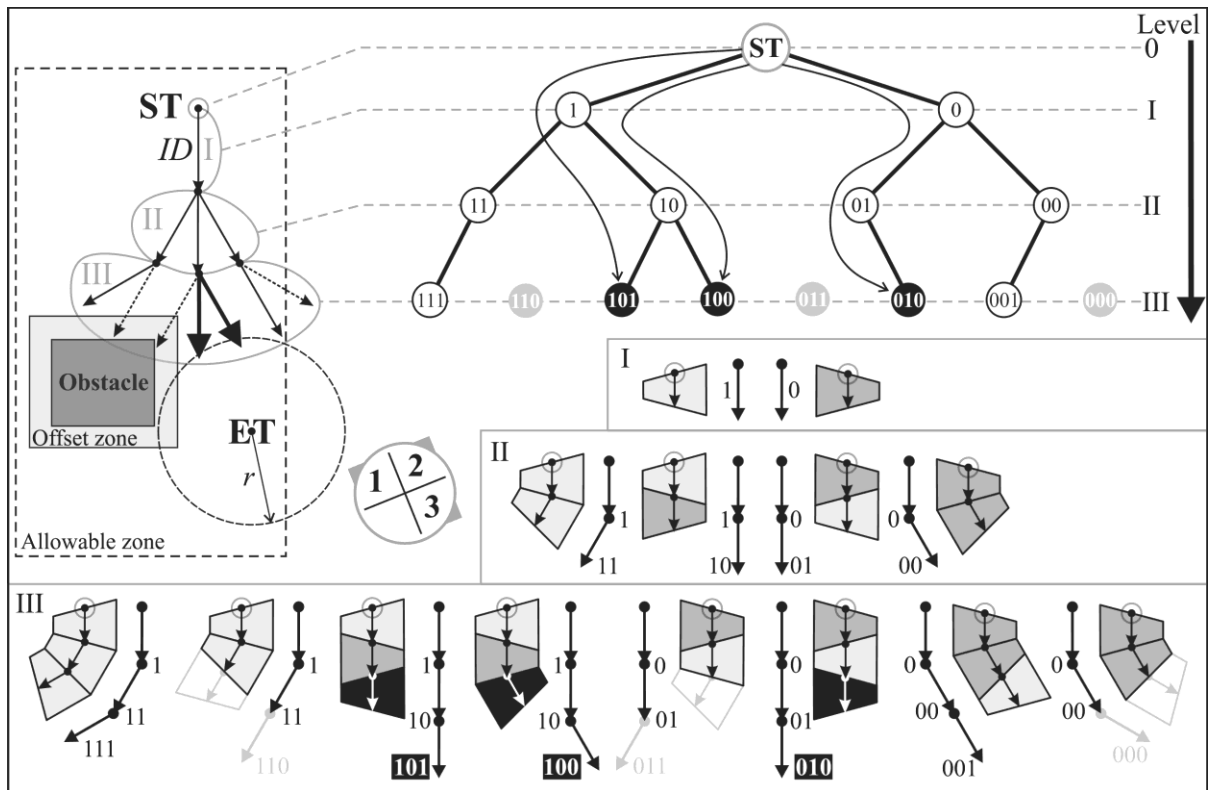
³ Rechenberg, I. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Dr.-Ing. Diss. Thesis, Technical University of Berlin, Department of Process Engineering, 1971.

Tab.1. Górny rząd: konstrukcje TZ: 1) złożona “ręcznie”, 2) wygenerowana przez *backtracking*, 3) najlepsza wygenerowana przypadkowo. Dolny rząd: metody ewolucyjne.

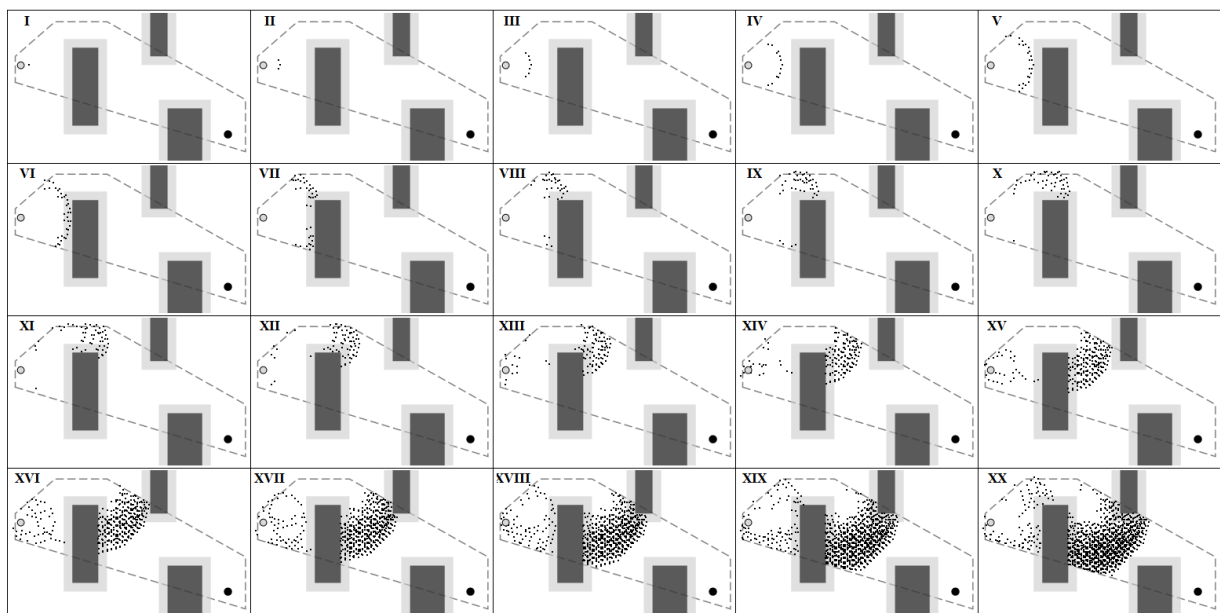


Trzeci artykuł w cyklu [4] wprowadza alternatywne sformułowanie zagadnienia tworzenia konstrukcji modularnej TZ przy użyciu metod grafowych, mianowicie jako tzw. *ścieżki* (ang. path graphs) TZP, co ilustruje Fig. 10.

Opublikowane wcześniej metody dawały dobre wyniki bez gwarancji, że są absolutnie najlepsze (optymalne). Celem niniejszego podejścia jest efektywne znajdowanie idealnych ścieżek TZ. W tym celu opracowano algorytmu, w którym TZ „eksploruje przestrzeń” tworząc drzewo binarne SST (nie oparte na siatce jak w przypadku algorytmu Dijkstry), co ilustruje Fig. 11.

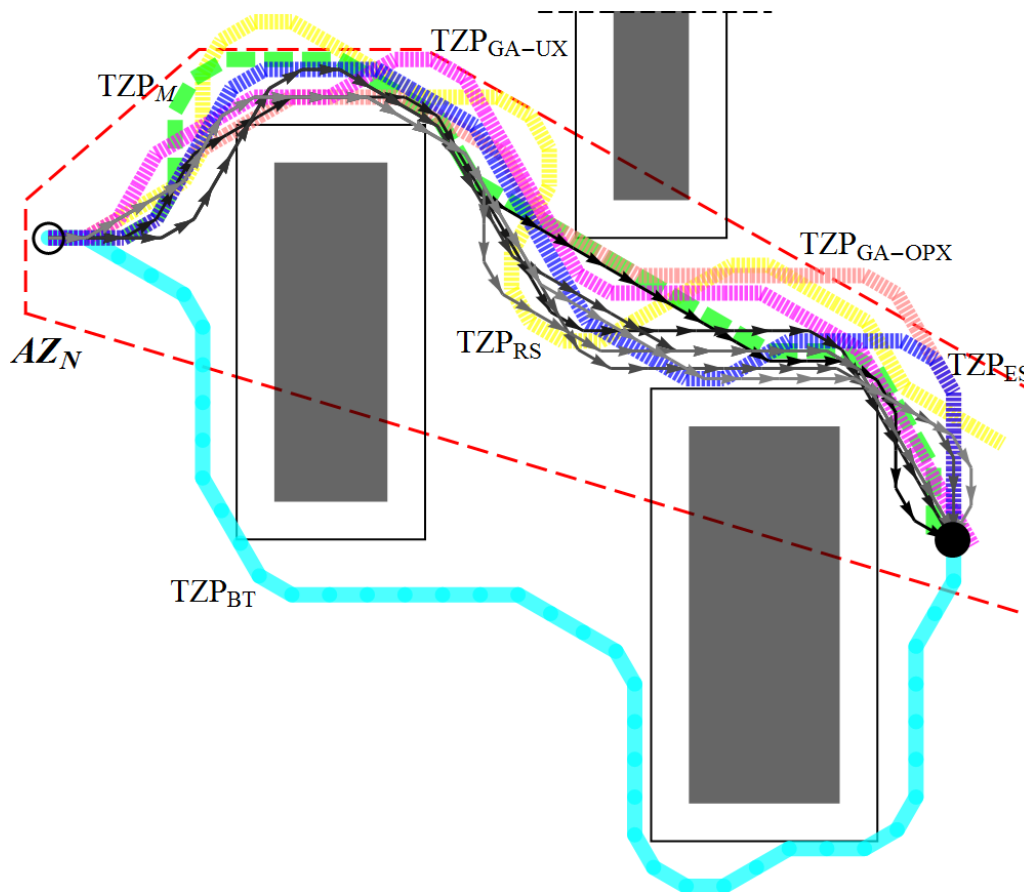


10. Przykład 3 alternatywnych ścieżek TZP. 1) „Framework” TZP w przestrzeni rzeczywistej. 2) Drzewo przeszukiwania SST przestrzeni ze wszystkimi dopuszczalnymi ścieżkami. 3) Wszystkie dopuszczalne konfiguracje TZP korespondujące z SST.



11. 20 kroków algorytmu korespondujące do 20 poziomów SST. Dla przejrzystości pokazano tylko wierzchołki generowane w danym kroku. Terminale ST i ET pokazano odpowiednio jako: jasnoszary i czarny punkt.

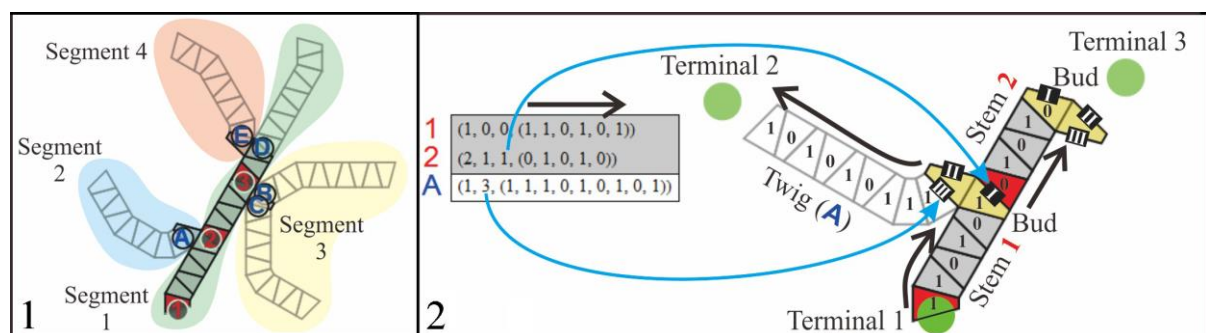
Fig. 12 zbiera wszystkie dotychczasowe wyniki.



12. M: rozwiązanie „ręczne”, GA-UX: algorytm genetyczny (uniform crossover), GA-OPX: algorytm genetyczny (one-point crossover), ES: evolution strategy, BT: backtracking. Strzałkami zaznaczono 6 idealnych rozwiązań grafowych.

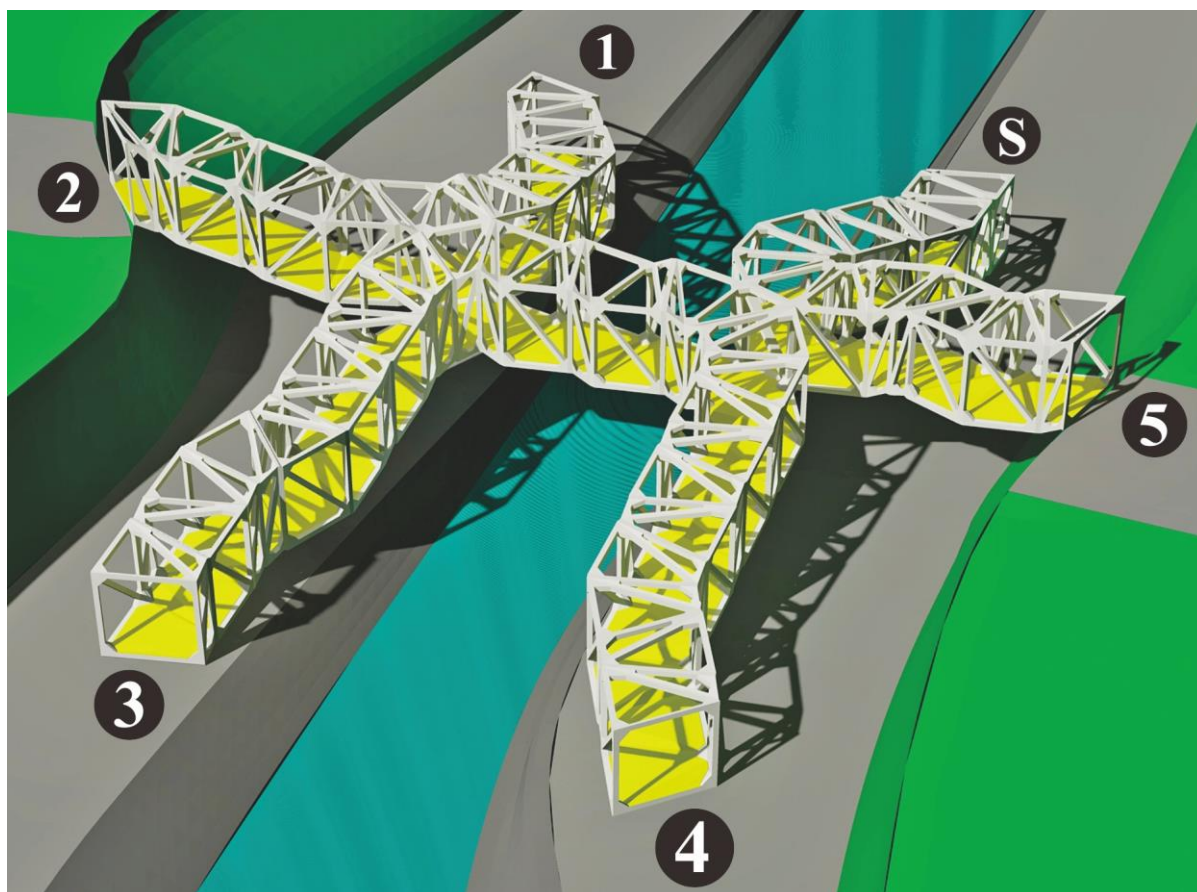
Dodatkowo artykuł wprowadza nowe kryteria do algorytmu optymalizacyjnego, mianowicie: „prostoty geometrycznej” oraz „liczby zakrętów”. Pracę wieńczy realistyczne studium przypadku instalacji TZ tworzącej pochylnię dla niepełnosprawnych ponad istniejącą drogą i zabudowaniami w bardzo ograniczonych (trudnych) warunkach geometrycznych. Opracowany algorytm wygenerował idealne rozwiązanie składające się ze 144 modułów, co ilustruje Fig.1.

Czwarty artykuł w cyklu [5] wprowadza formalny opis wielogłęziowych modularnych struktur Truss-Z (MTZ), co ilustruje Fig. 13.



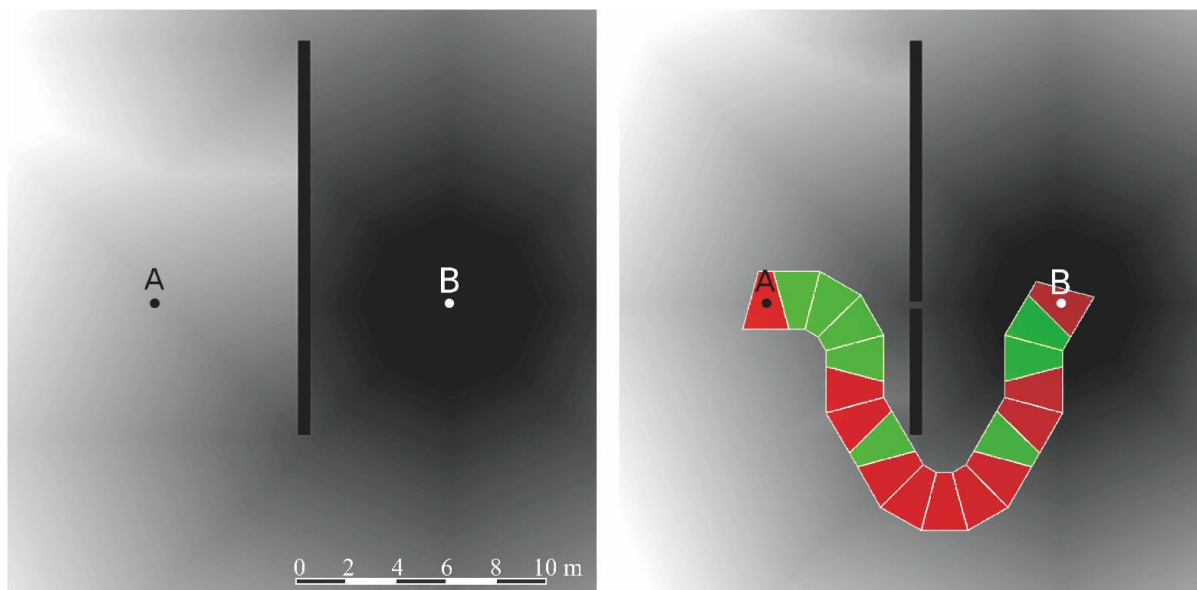
13. 1) Ilustracja pojęcia „segment”. 2) Nowa inspirowana Naturą nomenklatura.

Ponadto wprowadzono 6 nowych operatorów modyfikujących MTZ, funkcji dodatkowych: generowanie losowych struktur MTZ, naprawa zdegenerowanych struktur MTZ, transformacja liniowego genotypu w dwuwymiarową macierz, ilościowe porównywanie genotypów. Ponadto, wprowadzono 4 typy mutacji i wykazano, że w procesie optymalizacyjnym muszą być stosowane wszystkie. Pracę wieńczy realistyczne stadium przypadku instalacji wielogałęziowej modularnej konstrukcji mającej tworzyć sieć pochylni pieszko-rowerowych łączących 6 punktów w przestrzeni. Algorytm oparty o *strategię ewolucji* wygenerował dobre rozwiązanie tzn. złożone z 60 modułów TZ (minimalna liczba) i łączące zadane punkty z najmniejszym błędem. Fig. 14.



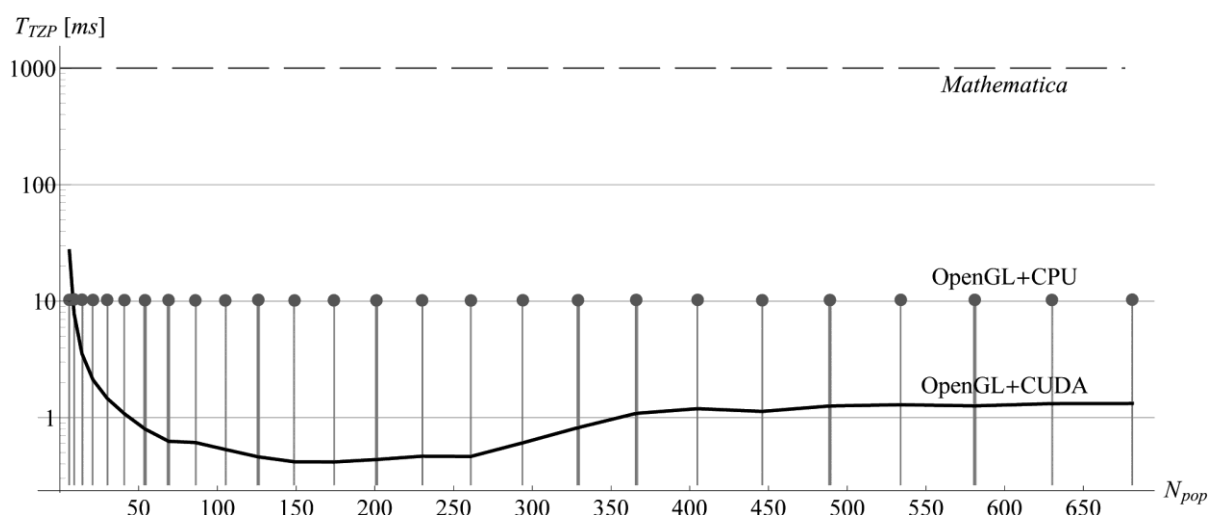
15. Wizualizacja komputerowa w widoku perspektywicznym MTZ łączącego Terminal S z 5-oma terminalami o różnej wysokości.

Piąty artykuł w cyklu [6] wprowadza metody analizy obrazu do oceny ilościowej wszelkiego typu kolizji: z elementami otoczenia oraz auto-kolizji konstrukcji TZ. Metoda ta umożliwia selektywny dobór wag w różnym stopniu oceniających kolizję z np. różnymi rodzajami przeszkód otoczenia (drzewo, zarośla, rzeka, budynek, grunt itp.). Opracowano metodę wyliczania funkcji celu bazującej na dowolnych macierzach kosztu oraz macierzach przepływu kosztu. Zaimplementowano algorytm genetyczny kodujący sekwencje TZ o zmiennej długości. Fig. 15 ilustruje skuteczność algorytmu w stosunkowo prostym, lecz obliczeniowo niebanalnym przypadku.



15. Skala szarości ilustruje „pole odległości” (ang. distance field) między punktami A i B. Po lewej pokazano prostą nieprzekraczalną przeszkodę. Po prawej – przeszkoda ta ma otwór, zbyt mały jednak by TZ mogło się w niej zmieścić. Niemniej algorytm jest skuteczny również w tej sytuacji – pokazano rozwiązanie idealne które powstało już po 15 krokach ewolucyjnych.

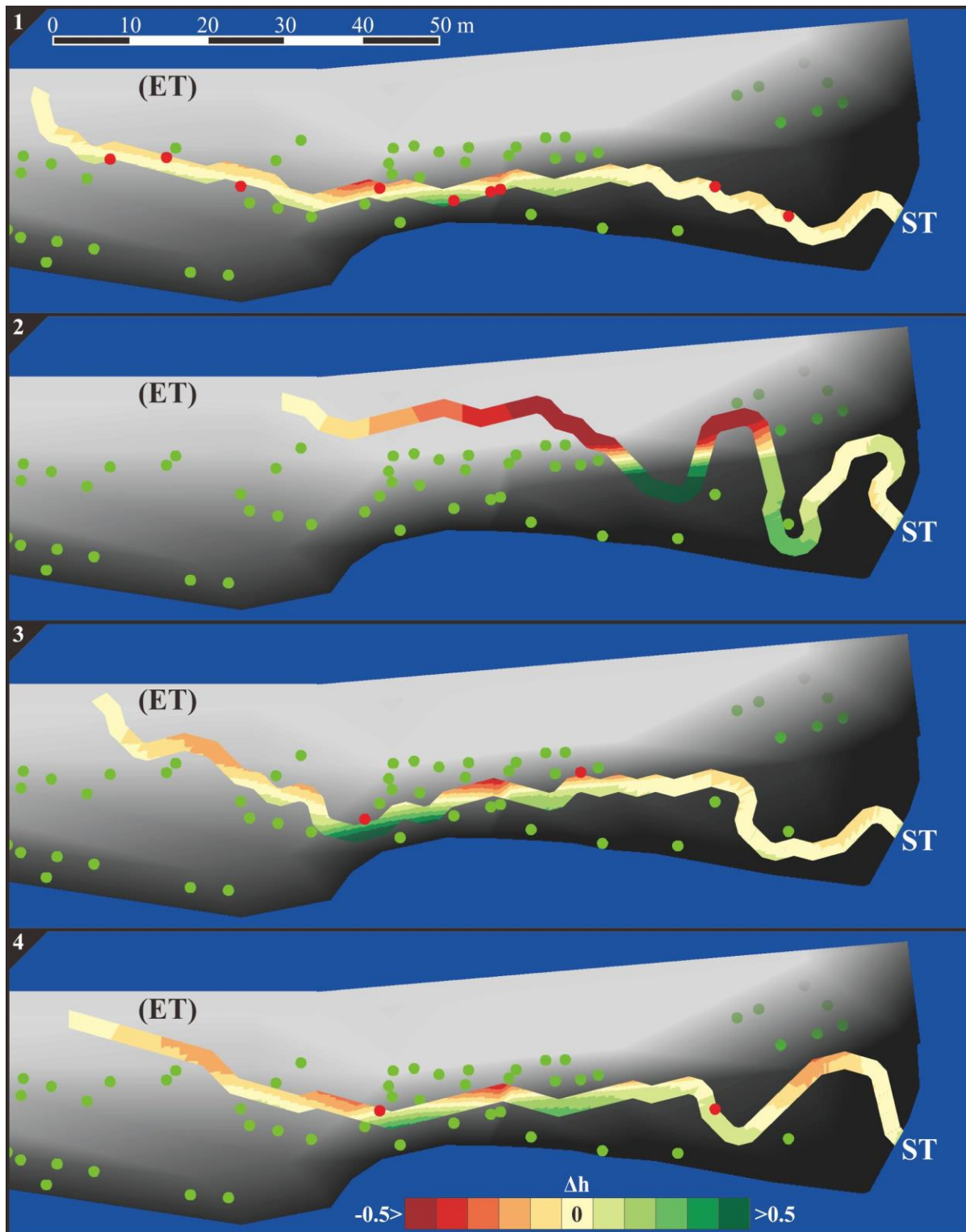
Przeprowadzono zrównoleglenie algorytmu w środowisku *Mathematica*TM na 4 fizycznych rdzeniach procesora oraz na platformie *Wolfram Lightweight Grid*TM. Następnie zaimplementowano algorytm w środowisku CUDA do wyliczania wartości funkcji celu w sposób wysoce efektywny na procesorach GPU. Fig. 16 ilustruje skalę przyspieszenia w porównaniu z niekompilowanym kodem w środowisku *Mathematica*TM. Jest to przyspieszenie rzędu 10^2 - 10^3 razy



16. Wykres logarymiczny zależności między czasem obliczeń t_u funkcji celu $u(S)$ dla pojedynczego genotypu S w populacji o liczebności N_{pop} .

Do algorytmu optymalizacyjnego wprowadzono dodatkowe kryteria takie jak koszty: robót ziemnych, usunięcia drzew i zarośli, przeprawy przez rzekę. Ponadto wprowadzono strefy zakazane, gdzie nie mogą być posadzone podpory TZ oraz implementacja wspomnianego wcześniej kryterium „prostoty geometrycznej”. Pracę wieńczą trzy studia przypadków o rosnącym stopniu realności: „modularna ścieżka w ogrodzie z rzeką i zaroślami”, „górski pomost” i instalacja pochylni dostępowej na stacji PKP Warszawa-Powisłe. Fig. 17 ilustruje

4 alternatywne rozwiązania dla ostatniego przypadku. Przewyższenie między ST i ET wynosi przeszło 10m, na stoku rośnie pewna liczba drzew. ET nie jest określone jako punkt tylko jako pewien obszar wzdłuż górnej części stoku.



17. Widok z góry czterech alternatywnych rozwiązań. 1) Minimalizacja robót ziemnych za cenę wycinki drzew. 2) Zachowanie wszystkich drzew. 3) Rozwiązanie optymalne – równoczesna minimalizacja robót ziemnych i wycinki drzew. 4) Rozwiązanie „optymalnie-proste” z dodatkowym kryterium „prostoty geometrycznej”. Legenda na dole ilustruje skalę robót ziemnych. Kolor czerwony i zielony oznaczają odpowiednio: wykopanie i usypanie ziemi o wysokości ponad 0.5 metra. Punkty: zielone i czerwone wskazują drzewa odpowiednio: istniejące i do wycinki.

Fig. 18 pokazuje fizyczny model trzeciego rozwiązania (optymalnego).



19. Fotografia fizycznego modelu rozwiązania optymalnego.

Dotychczasowe badania koncentrowały się optymalizacji sekwencji modułów lub wiązki takich sekwencji w przypadku konstrukcji wielogłęziowych. W takiej optymalizacji moduły są obiektami czysto geometrycznymi.

Szósty artykuł w cyklu [7] przedstawia wstępną *optymalizację strukturalną* modułu TZ. Optymalizowane są równocześnie: topologia modułu (układ belek przekątnych) oraz wymiary przekrojów elementów. Ponieważ system TZ ma być uniwersalny, przedmiotem optymalizacji nie jest jedna konstrukcja, ale wszystkie zasadniczo unikalne konfiguracje o liczbie modułów $n = 2, \dots, 7$. Tego typu sformułowanie odpowiada optymalizacji „najgorszej” konfiguracji i w naturalny sposób pozwala na zrównoleglenie obliczeń. Fig. 19 ilustruje 16 „najgorszych” 7-mio modułowych konfiguracji TZ.

Funkcją celu jest masa pojedynczego modułu $m(\mathbf{x}, d)$, która bezpośrednio zależy od wektora \mathbf{x} przekrojów belek oraz topologicznej konfiguracji belek przekątnych d . Ograniczenia, poza oczywistymi dolnymi ograniczeniami wymiarów belek, zadane są w postaci górnego ograniczenia naprężenia efektywnego σ^{eq} w elementach konstrukcji:

$$\sigma^{eq}(\mathbf{x}, d, S_n) := \max_{s \in S_n} \sigma^{eq}(\mathbf{x}, d, s) \leq 100 \text{ MPa} \quad (7)$$

gdzie s jest pojedynczą globalną konfiguracją konstrukcji modularnej, podczas gdy S_n jest zbiorem rozpatrywanych konfiguracji. Zmienne optymalizacyjne są oznaczone d (konfiguracja topologii belek przekątnych) i \mathbf{x} (wektor wymiarów przekrojów belek). Wartość 100 MPa jest wartością przyjętą arbitralnie i uznaną za bezpieczną dla stali.

1 {4, 5}	2 {1}	3 {4, 5}	4 {1}
5 {3, 4}	6 {1}	7 {3, 4}	8 {7}
9 {7}	10 {6}	11 {6}	12 {7}
13 {7}	14 {1}	15 {1}	16 {1}

19. Dla każdego przypadku pokazano indeks najbardziej wyťažonej belki w module oraz odpowiednie indeksy modułów. Belka ta jest również zaznaczona na czerwono.

Ponieważ zmienne optymalizacyjne są dwóch rodzajów, proces optymalizacji jest prowadzony dwupoziomowo: na poziomie "wyższym" optymalizowana jest topologia belek przekątnych. Dla każdej rozważanej topologii przeprowadzona jest w sposób niezależny optymalizacja poziomu „niższego”, tzn. względem przekrojów belek modułu, patrz Tab. 2.

$$\bar{\mathbf{x}}(d, S_n) := \arg \min_{\mathbf{x}} m(\mathbf{x}, d) \quad (8)$$

gdzie $\bar{\mathbf{x}}(d, S_n)$ jest wektorem optymalnych przekrojów belek dla topologii d i zbioru konfiguracji S_n .

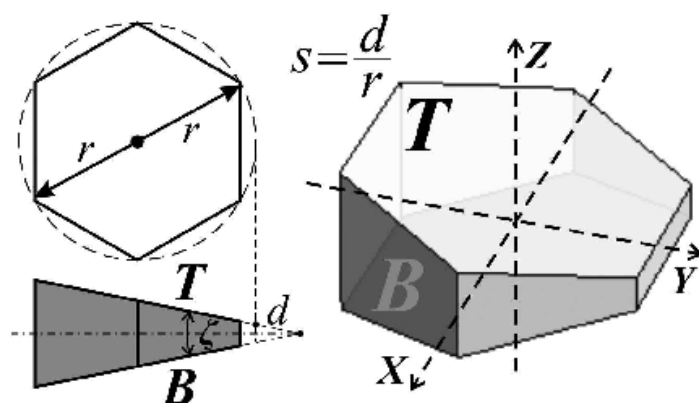
Optymalizacja topologii d ma charakter dyskretny i - ze względu na niewielką liczbę konfiguracji przekątnych - przeprowadzona jest metodą wyszukiwania wyczerpującego. Optymalizacja poziomu niższego (względem wektora przekrojów belek \mathbf{x}) przeprowadzona jest metodą lokalnej ewolucji z adaptacyjnie zmienną definicją otoczenia: otoczenie jest dynamicznie modyfikowane w trakcie optymalizacji w taki sposób, by w miarę postępu procesu zintensyfikować przeszukiwanie w otoczeniu aktywnych ograniczeń.

Tab.2 Ogólny schemat procedury optymalizacyjnej.

<p>For each considered set $S_n \in S$:</p> <ol style="list-style-type: none"> 1. Select the initial vector $\bar{\mathbf{x}}$ of the beam diameters. 2. For each configuration \mathbf{d} of the diagonals: <ol style="list-style-type: none"> a. Compute the mass $m(\mathbf{x}, \mathbf{d})$ of the module. b. Compute the local stiffness matrix and the local load vector of a single module. c. For each configuration $\mathbf{s} \in S_n$ <ol style="list-style-type: none"> i. Aggregate the local stiffness matrix and the local load vector of single modules into the global stiffness matrix and the global load vector of the configuration \mathbf{s}. ii. Compute the global displacement vector and the compliance $C(\mathbf{x}, \mathbf{d}, \mathbf{s})$. iii. For each beam, compute the local nodal reaction forces and use the circular hollow section formulas to compute the distribution of the von Mises stress and its maximum value. Store the maximum values, as found for each beam of the module, in the vector $\sigma^{\text{eq}}(\mathbf{x}, \mathbf{d}, \mathbf{s})$. d. Take the maximum of $C(\mathbf{x}, \mathbf{d}, \mathbf{s})$ w.r.t. all $\mathbf{s} \in S_n$ to find the maximum (worst-case) compliance $C(\mathbf{x}, \mathbf{d}, S_n)$. e. Take the maximum of the successive elements of $\sigma^{\text{eq}}(\mathbf{x}, \mathbf{d}, \mathbf{s})$ w.r.t. all $\mathbf{s} \in S_n$ to find the vector $\sigma^{\text{eq}}(\mathbf{x}, \mathbf{d}, S_n)$ of the maximum von Mises stresses of the beams. f. Check the stress constraints. If not satisfied, update $\bar{\mathbf{x}}$ and go to Point 2a. g. Check the stop conditions. If not satisfied, update $\bar{\mathbf{x}}$ and go to Point 2a. h. Store the optimum $\bar{\mathbf{x}}(\mathbf{d}, S_n)$, the corresponding mass of the module $m(\mathbf{d}, S_n)$, the worst-case compliance $\bar{C}(\mathbf{d}, S_n)$ and the maximum von Mises stresses of the beams $\sigma^{\text{eq}}(\bar{\mathbf{x}}, \mathbf{d}, S_n)$.
--

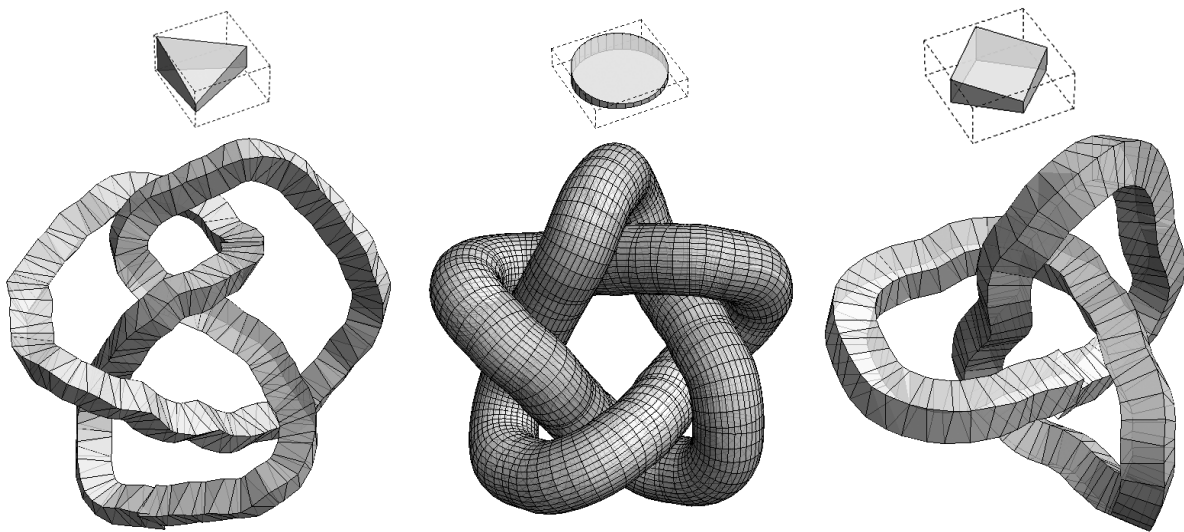
Powyższe prace dotyczyły systemu Truss-Z.

Siódmy artykuł w cyklu [8] dotyczy drugiego chronologicznie Systemu Ekstremalnie Modularnego, mianowicie – Pipe-Z, a dokładnie Arm-Z będącego jego dalszym rozwinięciem. Pipe-Z to modułowy system do tworzenia jednogłęziowych trójwymiarowych struktur rurowych o dowolnym kształcie z jednego rodzaju modułu o przekroju wielokąta. Moduł Pipe-Z określa kilka parametrów: liczba kątów k , rozmiar r , odsunięcie d , i kąt ζ między powierzchniami dolną (B) i górną (T). Wygodnie jest używać dodatkowego parametru $s = d/r$. Fig. 20 ilustruje te parametry.



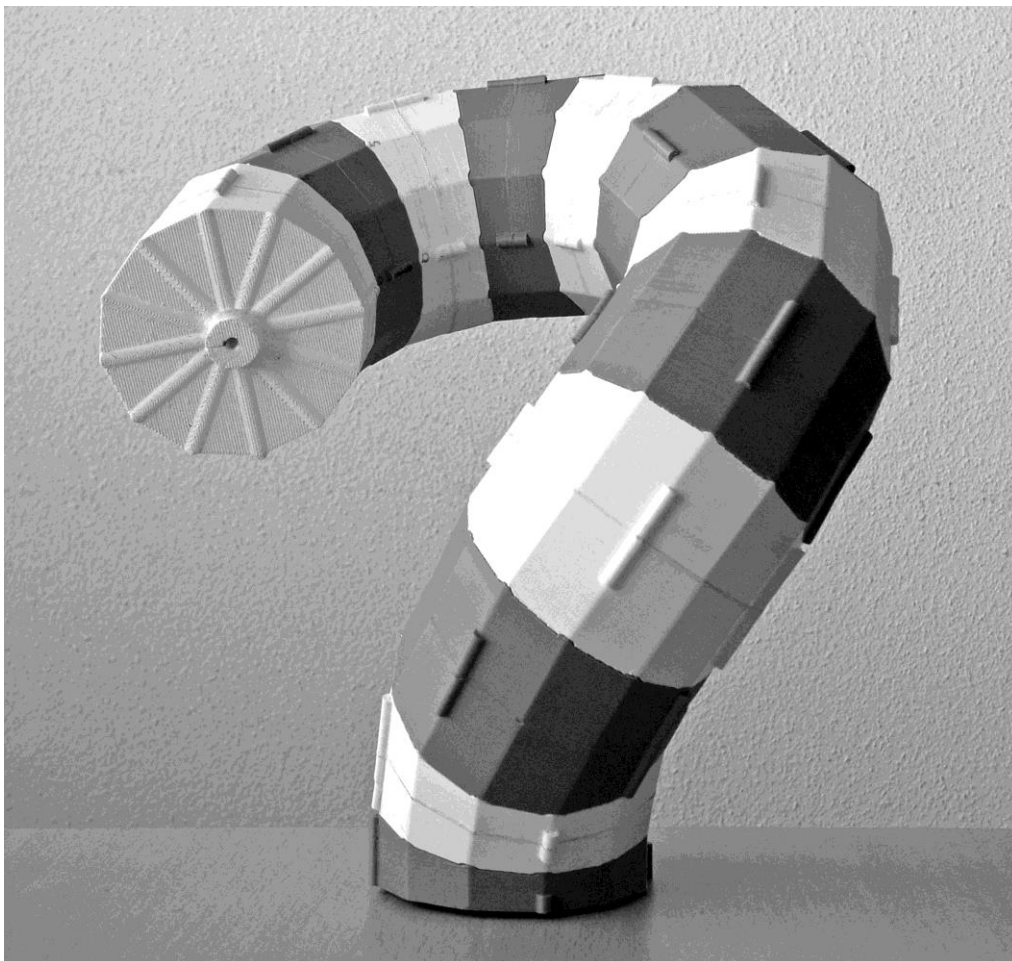
20. Wizualizacja sześciokątnego modułu Pipe-Z ($k = 6$). Po lewej od góry: widok z góry i widok z boku. Po prawej: widok aksonometryczny.

Fig. 21 pokazuje trzy przykłady Pipe-Z.



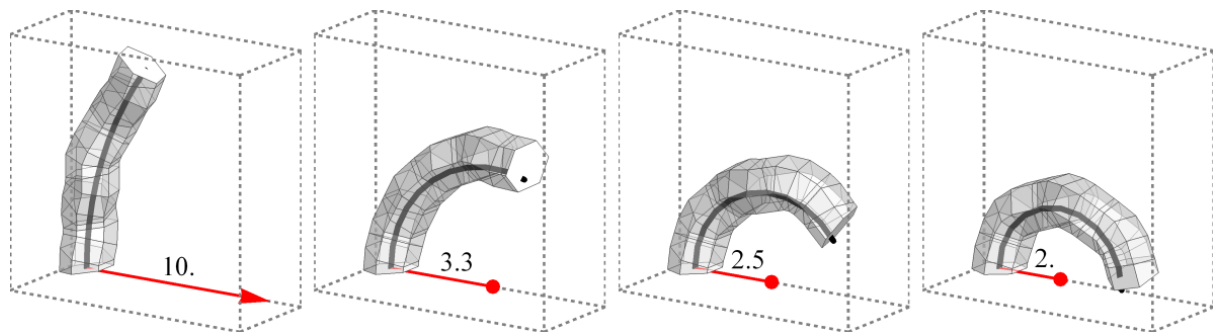
21. Trzy przykłady Pipe-Z wraz z modułami, z których są skonstruowane. Od lewej: „ósemka” (4₁), *cinquefoil* (5₁) i *trefoil* (3₁).

Arm-Z to manipulator typu „trąba słonia” zbudowany na podstawie Pipe-Z, w którym pierwszy moduł jest utwierdzony a kolejne posiadają jeden stopień swobody (1DOF). Manipulator taki ma tyle stopni swobody, ile modułów, przeto w literaturze klasyfikowany jest jako „hiperredundantny”. Fig. 22 pokazuje fotografię wstępnego prototypu Arm-Z.



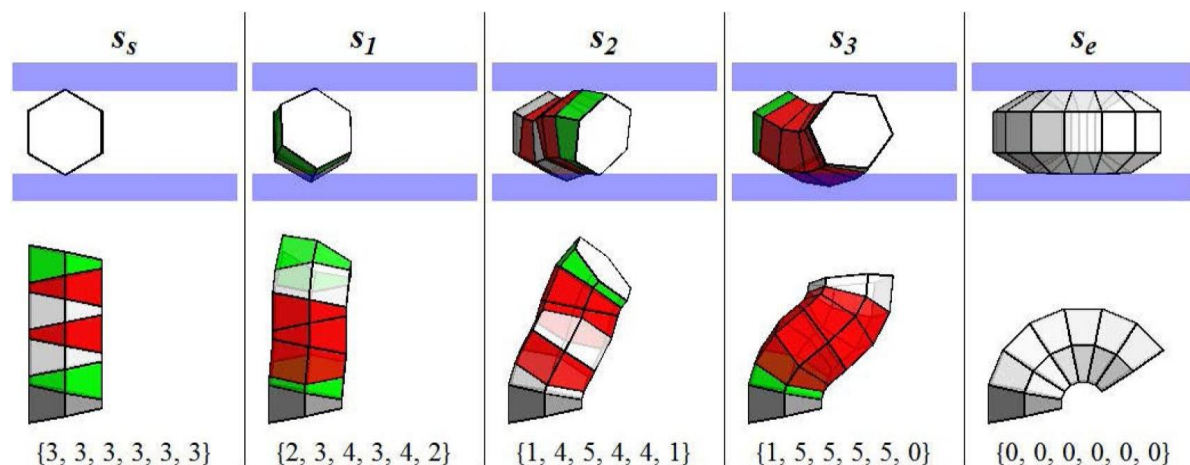
22. Fotografia prototypu Arm-Z zbudowanego z 12-u 12-kątnych modułów.

Analogicznie do opisanego wyżej łączenia modułów TZ wzdłuż danej krzywej w przestrzeni (GP), Arm-Z (w skrócie AZ) może być kontrolowane bezpośrednio przez dynamicznie zmieniającą się taką krzywą. Fig. 23 ilustruje implementację algorytmu zachłannego dopasowującego sześciokątny 16-to modułowy AZ do parametrycznej krzywej zmieniającej się z pionowej linii prostej do połówki łuku kołowego.



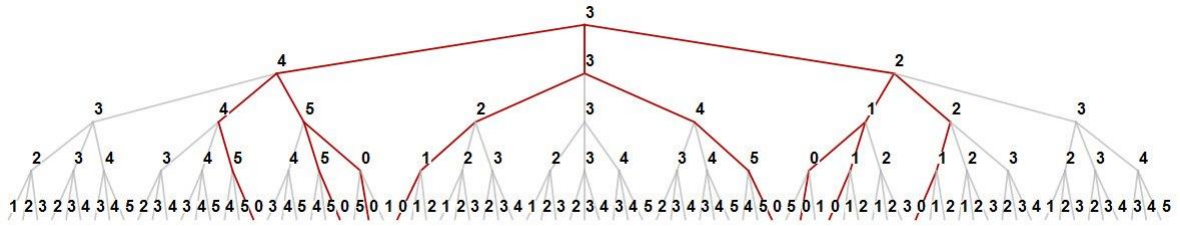
23. Cztery wybrane stany z ciągłego dopasowywania AZ do zmieniającej się krzywej GP (pokazanej grubą linią wewnątrz AZ). Długość promienia krzywej GP pokazano dla każdego przypadku.

Problem transformacji AZ został postawiony bardziej fundamentalnie: znajdź najkrótszą transformację ze stanu początkowego S_s do stanu końcowego S_e w szczeliny pionowej. Dodatkowo ograniczono zakres obrotów każdego modułu do jednego „kliknięcia” w dowolną stronę w trakcie jednego kroku tej dyskretnej transformacji. Znając S_s i S_e , można bardzo łatwo znaleźć bezpośrednią transformację po prostu rozpatrując oddzielnie każdy moduł. Niemniej rozwiązanie takie powoduje bardzo chaotyczny ruch manipulatora i w efekcie niedopuszczalne kolizje ze szczeliną. Zaimplementowano zatem algorytm grafowego wyszukiwania wyczerpującego. Dla sześciokątnego PZ wszystkie moduły oprócz pierwszego mogą obracać się o 6 dyskretnych kątów: $\{0, \pi/3, 2\pi/3, \pi, 4\pi/3, 5\pi/3\}$, którym odpowiadają liczby: $\{0,1,2,3,4,5\}$. Konfiguracja $\{3,3,3,3,3,3\}$ oznacza ustawienie naprzemienne dające „prosty pręt”, zaś $\{0,0,0,0,0,0\}$ tworzy łuk. Fig. 24 ilustruje idealne rozwiązanie dla przejścia z „pręta prostego” do łuku dla sześciokątnego 7-o modułowego AZ .



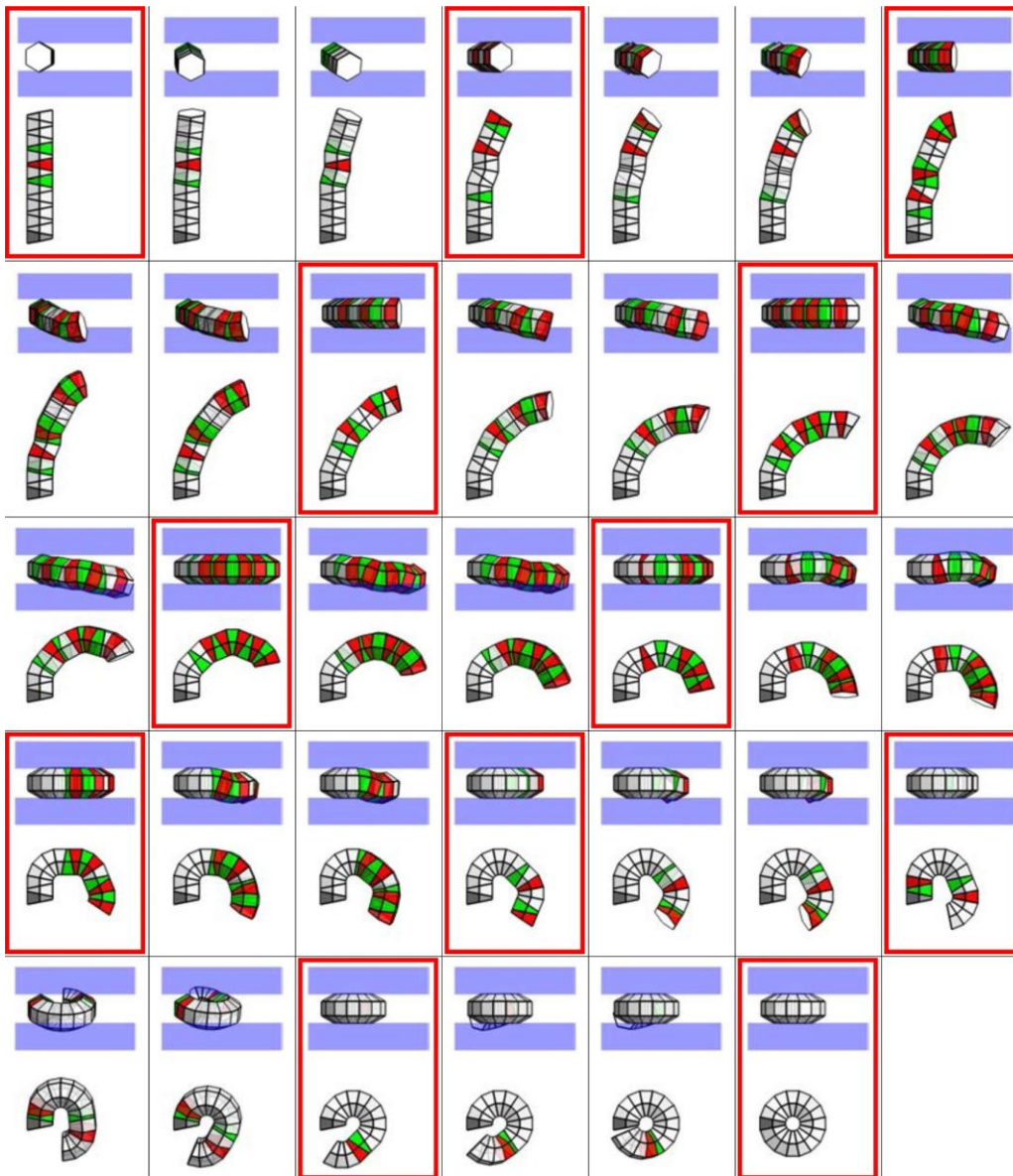
24. Idealna (najkrótsza dopuszczalna) transformacja sześciokątnego 7-mio modułowego AZ . Pierwszy moduł (zaznaczony na szaro) jest utwierdzony. Kolory czerwony i zielony wskazują, że w następnym kroku dany moduł skręci odpowiednio w: lewo lub prawo. Niebieskie paski wyznaczają szerokość szczeliny.

Ponieważ w każdym kroku każdy moduł może przybrać 3 pozycje: skręt w prawo, skręt w lewo lub pozostać bez zmiany, drzewo potencjalnych ścieżek ma formę drzewa trójkowego, co ilustruje Fig. 25.



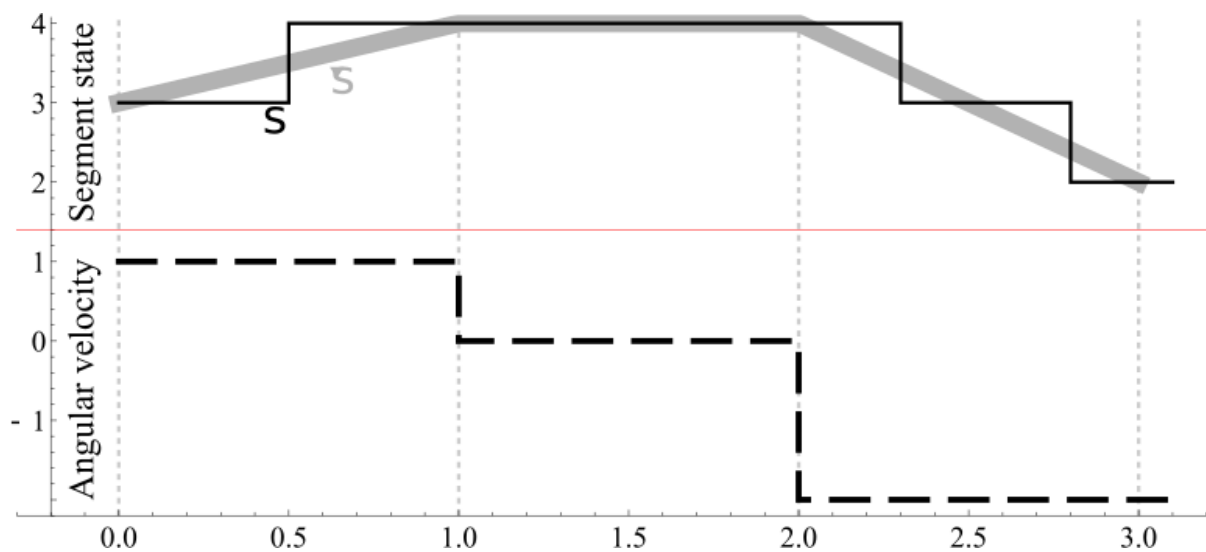
25. Dla 4 kroków każdy moduł może przejść ze stanu 3 do stanu 0 na 8 sposobów.

Zatem liczba wszystkich możliwych transformacji ze stanu $S_s (3,3,3,3,3,3)$ do $S_e (0,0,0,0,0,0)$ wynosi $8^6 = 262144$. Sprawdzenie ponad ćwierć miliona konfiguracji pod kątem kolizji ze szczeliną zajmuje ok. 4h na standardowym komputerze. Jak widać liczba wszystkich możliwości rośnie podwójnie wykładniczo wraz z liczbą modułów. Oznacza to, że jest to metoda stosowalna wyłącznie do bardzo małych przypadków. Następnie zaimplementowano algorytm zachłanny, który wygenerował pewne rozwiązanie dla sześciokątnego 16-to modułowego AZ, co ilustruje Fig. 26.



26. Transformacja wygenerowana przez algorytm zachłanny potrzebuje 33 kroków. Co 3 kroki idealnie „prostuje się” w szczelinie, co zaznaczono czerwoną ramką.

Algorytmy zachłanne generują zwykle dopuszczalne rozwiązania w rozsądnym czasie, niemniej na ogół nie są to rozwiązania optymalne. Dla tego przypadku (16 modułów) liczba czterokrokowych transformacji wynosi $8^{15} = 35,184,372,088,832$. Jest to liczba astronomiczna i wyszukiwanie wyczerpujące jest nierealne. Ponadto nie ma powodu ograniczać transformacji każdego z modułów tylko do czterech kroków (jak pokazuje Fig.26). Tym bardziej stosowanie metod brute-force w tym przypadku nie jest racjonalne. Zatem w celu polepszenia jakości rozwiązań zaimplementowano optymalizację rojem cząstek (PSO). Metoda ta jest adekwatna do tego bardzo nieliniowego problemu optymalizacyjnego, gdyż umożliwia przeszukiwanie bardzo dużego i nieregularnego obszaru rozwiązań. W tym celu „uciąglono” zmiany położenia modułów względem siebie, jak ilustruje Fig. 27.



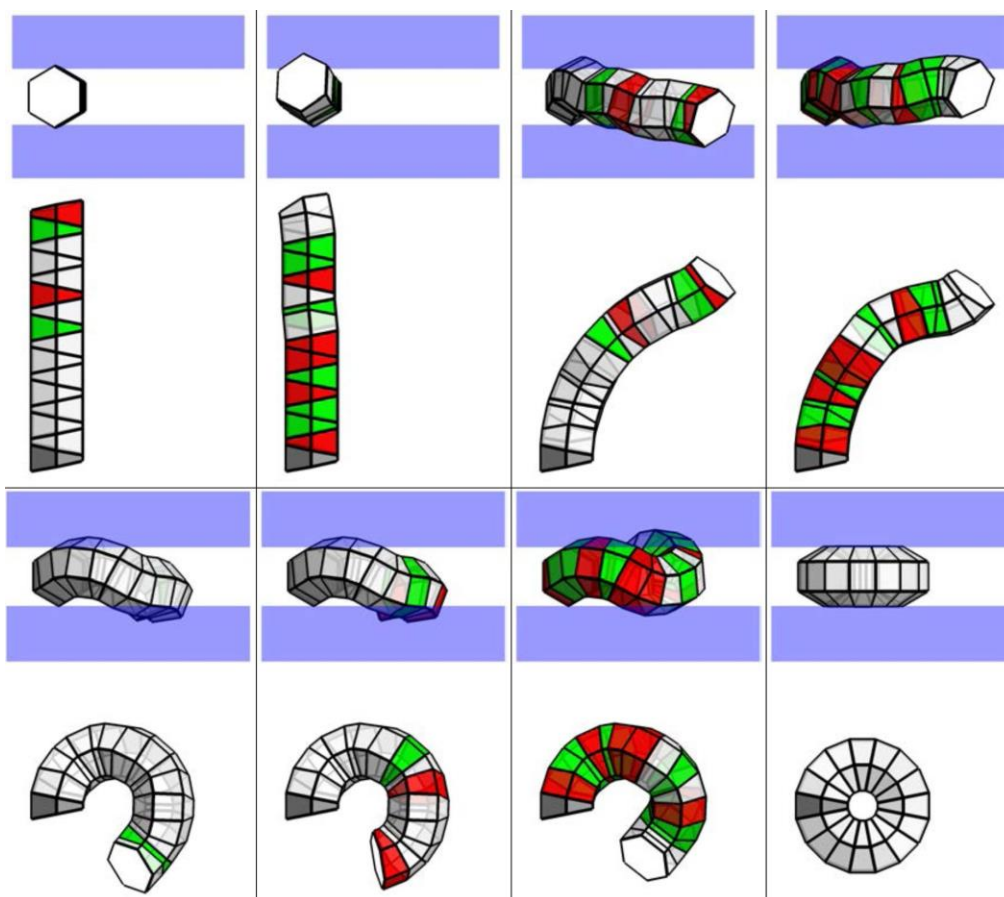
27. Przykładowy wykres ilustrujący „uciąglenie” dyskretnych zmiany położenia modułu w czasie.

Tab.3. Pokazuje schemat procedury optymalizacyjnej PSO.

Tab.3. Schemat procedury optymalizacyjnej rojem cząstek PSO. Każda cząstka reprezentuje przejście prędkości Ω czyli $l \times k$ liczb rzeczywistych z zakresu $[-1, 1]$. $\mu_{\min} = -1$, $\mu_{\max} = 1$, $\eta_{\min} = -2$, $\eta_{\max} = 2$.

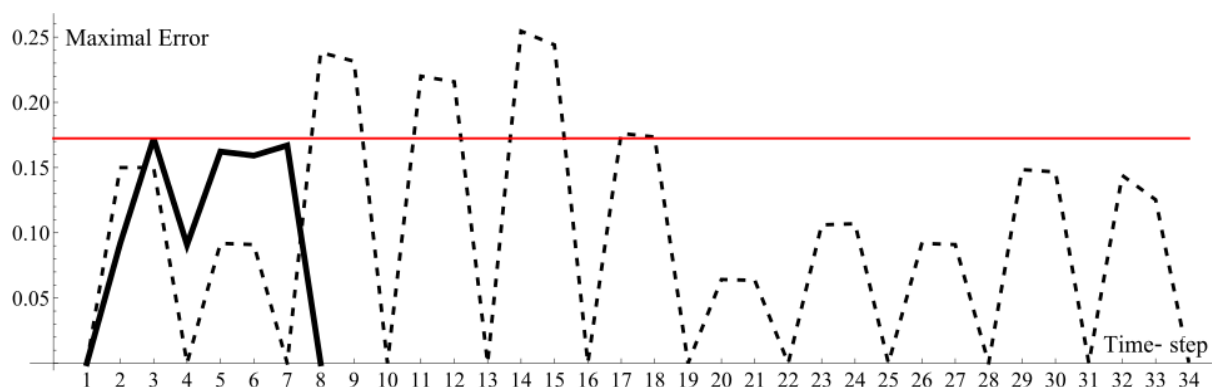
1	For each particle i :
2	initialize \mathbf{x}_i with random values, $\mu_{\min} \leq x_i, j \leq \mu_{\max}$
3	let the particle's best position \mathbf{p}_i be equal to its initial position: $\mathbf{p}_i \leftarrow \mathbf{x}_i$
4	initialize \mathbf{v}_i with random values, $\eta_{\min} \leq v_i, j \leq \eta_{\max}$
5	calculate the objective function for particle i : $f_i(\mathbf{p}_i)$
6	update the swarm best solution: if $f_i \leq f(\mathbf{b})$ then $\mathbf{b} \leftarrow \mathbf{x}_i$
7	For each particle i :
8	select random numbers r_p, r_b from the range $[0, \dots, 1]$
9	for each component j :
10	$v_{i,j} \leftarrow \omega_g v_{i,j} + \phi_p r_p (p_{i,j} - x_{i,j}) + \phi_b r_b (b_j - x_{i,j})$
11	$\mathbf{x}_i \leftarrow \mathbf{x}_i + \mathbf{v}_i$
11	if $f(\mathbf{x}_i) < f(\mathbf{p}_i)$ then the particle's best $\mathbf{p}_i \leftarrow \mathbf{x}_i$
12	update the swarm best solution: if $f_i \leq f(\mathbf{b})$ then $\mathbf{b} \leftarrow \mathbf{x}_i$
13	If a given termination criterion is not met, go to 7

Fig. 28 pokazuje wynik optymalizacji rojem cząstek PSO dla tego problemu.



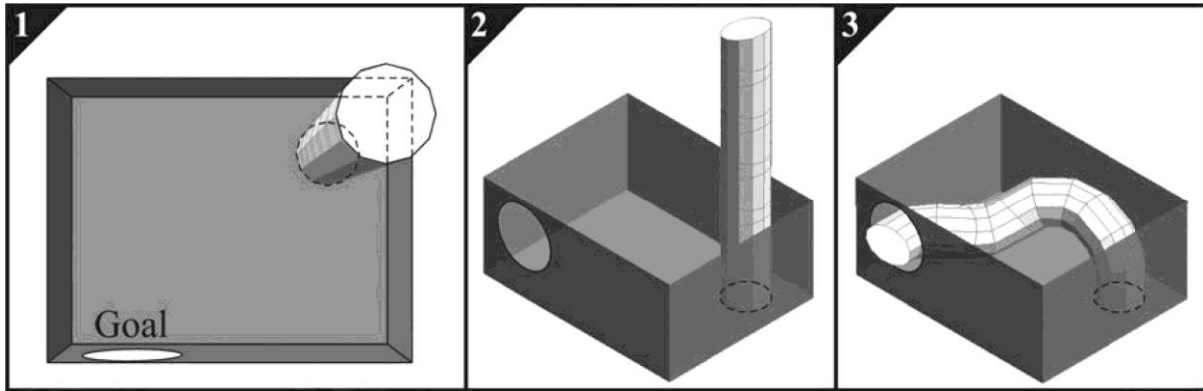
28. Najlepszy wynik PSO dla transformacji sześciokątnego 16-sto modułowego AZ.

Jak wskazuje Fig. 28, transformacja znaleziona przez PSO wymaga tylko 7 kroków. Fig. 29 porównuje błędy (stopień kolizji ze szczeliną) między rozwiązaniem PSO a „zachłannym”.



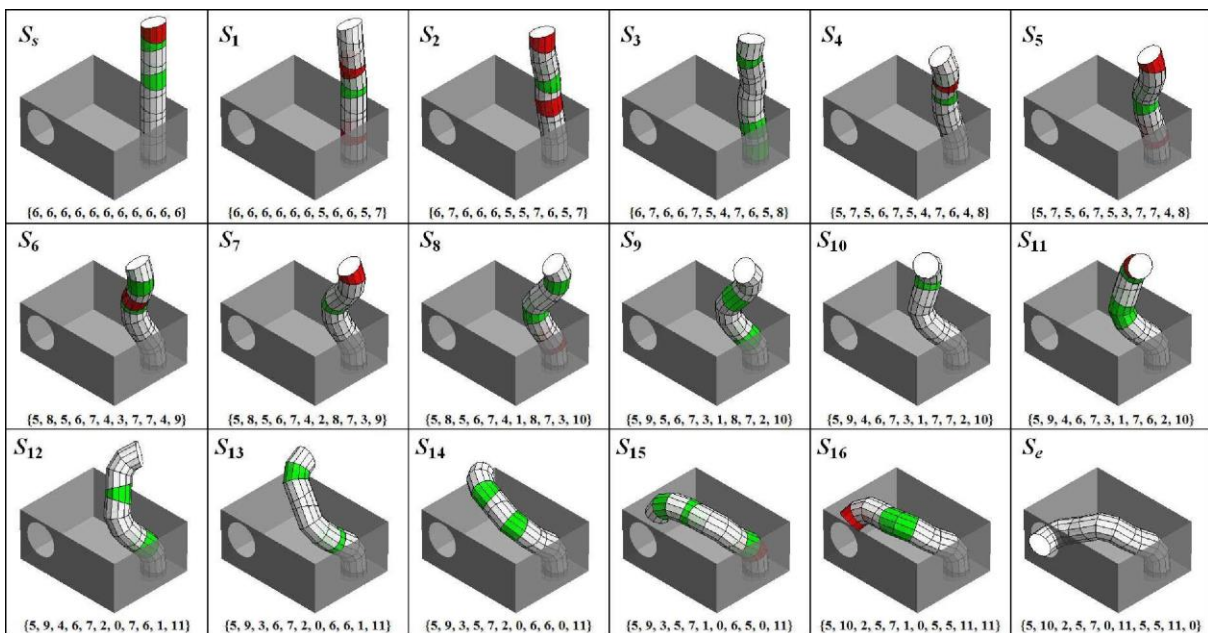
29. Porównanie maksymalnych błędów między transformacją wygenerowaną przez algorytm: zachłanny (linia przerywana) i PSO (linia ciągła).

Jak wskazuje Fig. 29, PSO znalazło transformację nie tylko dużo krótszą, ale i mniej „chybotliwą” czyli mniej kolidującą ze szczeliną. Artykuł więczą dwa przykłady transformacji w oparciu o następujący eksperyment: w zadanym środowisku przeprowadź transformację AZ ze stanu S_s do S_e bez kolidowania z tymże środowiskiem. Fig. 30 ilustruje założenia tego eksperymentu.



30. 1) Widok z góry konfiguracji startowej S_s w zadanym środowisku. 2) Widok aksonometryczny S_s . 3) Widok aksonometryczny S_e .

Dla tego już bardziej realistycznego problemu algorytm oparty o PSO wygenerował dobre rozwiązanie w ciągu kilku minut zaś bardzo dobre w ciągu ok. 3.5 h. Jeszcze ciekawszym (realistycznym) zagadnieniem jest postawienie problemu w ten sposób: konfiguracja końcowa S_e nie jest znana; przeprowadź końcówkę manipulatora AZ do otworu. Fig. 31 pokazuje transformację TZ dla tego problemu znalezioną przez PSO.

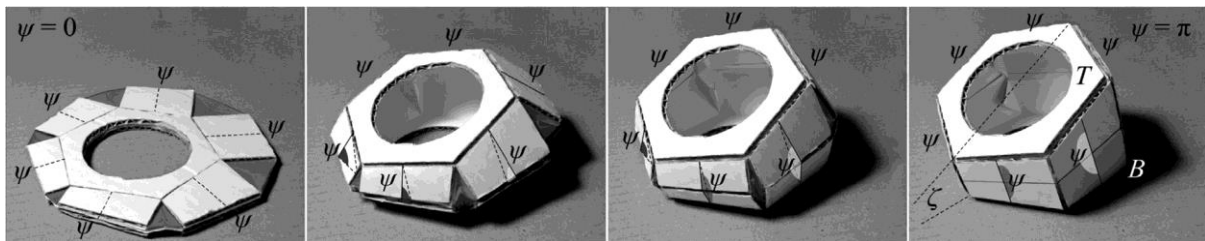


31. Najlepsze rozwiązanie znalezione przez PSO dla tego problemu potrzebuje tylko 17 kroków.

5. Omówienie pozostałych osiągnięć naukowo - badawczych.

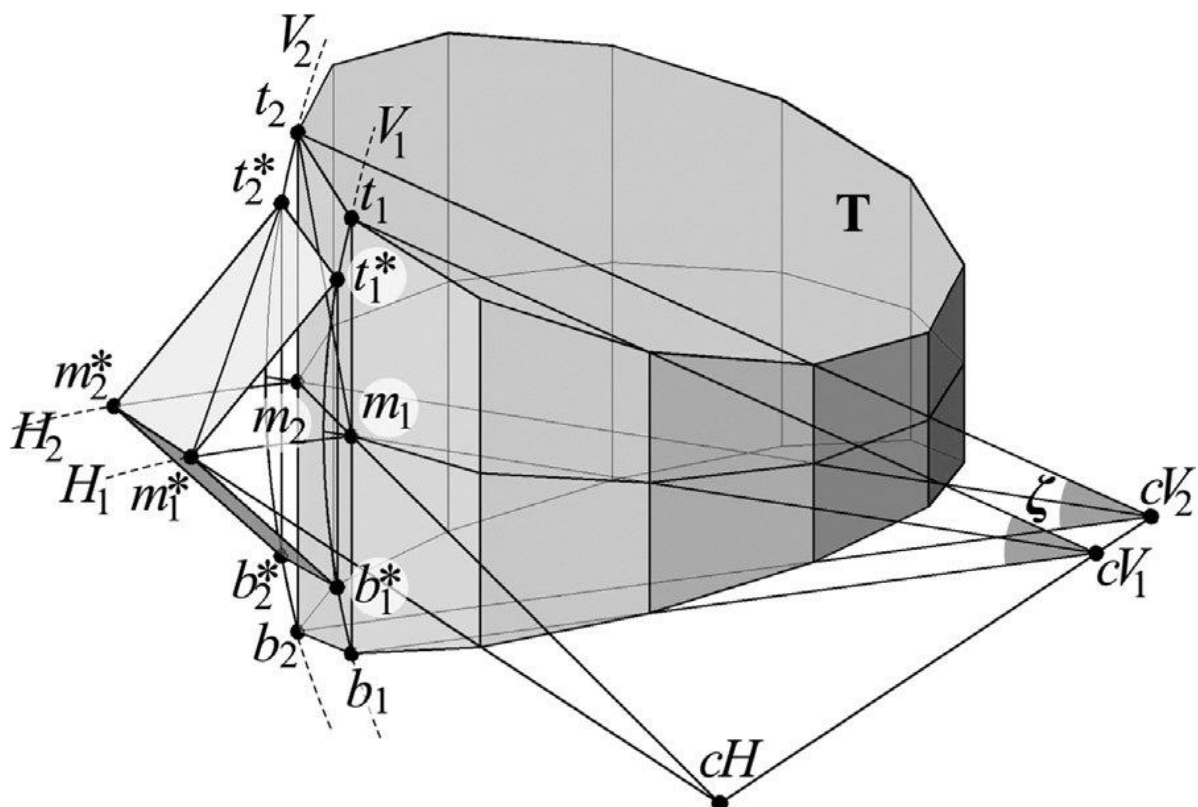
Warto wspomnieć o artykule, który nie znalazł się w cyklu habilitacyjnym, gdyż czasopismo należy do dyscypliny Inżynieria Materiałowa/Mechaniczna, ale jest odnośny, gdyż traktuje o systemie *Pipe-Z*, mianowicie: M. Zawidzki, [Deployable Pipe-Z](#), 2016, **ACTA ASTRONAUTICA**, 127, pp. 20–30. **IF5:1.6, Q1** (Engineering, Aerospace) **Lista A MNiSW** – 35 pkt.

Praca ta wprowadza zagadnienie składania modułów *Pipe-Z*, co ilustruje Fig. 32.



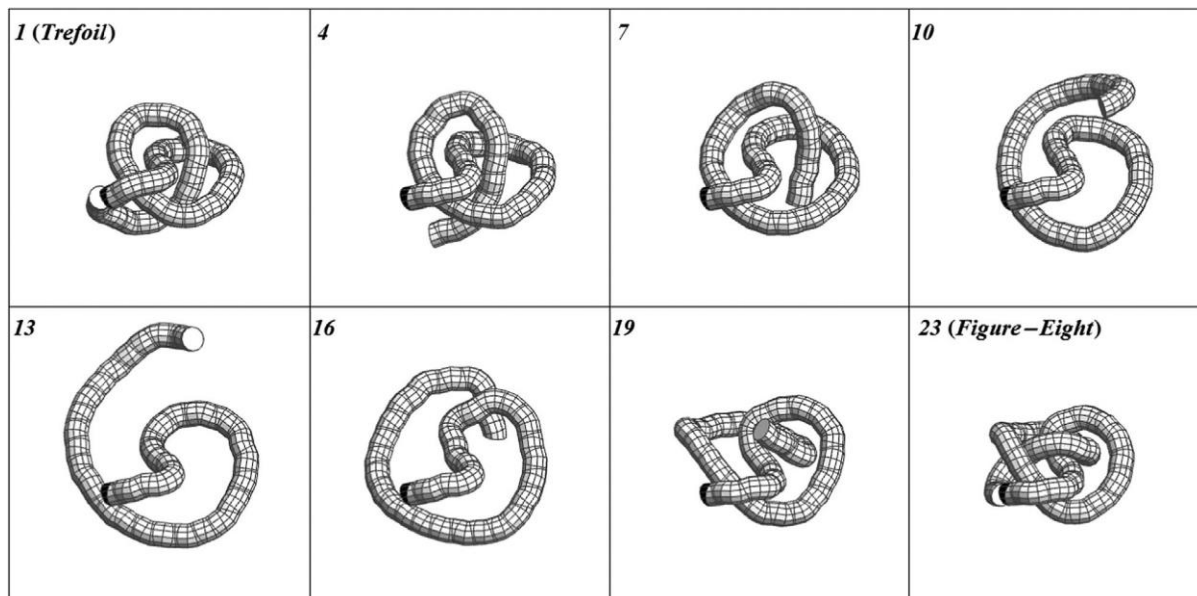
32. Cztery wybrane stany rozkładania modułu fPZ (ang. foldable Pipe-Z) od złożenia do kompletnego rozłożenia.

W artykule przeprowadzono również analizę geometryczną dowodzącą, że składanie takie nie powoduje odkształceń, co ilustruje Fig.33.



33. Analiza wykazująca niezmiennosć odległości w trakcie transformacji (“składania”) modułu, np.: $t_1 t_2 = t_1^* t_2^*$, $m_1 m_2 = m_1^* m_2^*$, $b_1 b_2 = b_1^* b_2^*$, $t_2 m_1 = t_2^* m_1^*$, $t_1 m_1 = t_1^* m_1^*$, itd.

Z algorytmicznego punktu widzenia, ciekawa jest przedstawiona na wstępie bezkolizyjna transformacja jednego modularnego matematycznego węzła - Trefoil (3_1) w inny – Ósemkę (4_1), co ilustruje Fig. 34.



34. Wybrane kroki z bezkolizyjnej transformacji jednego węzła w drugi.

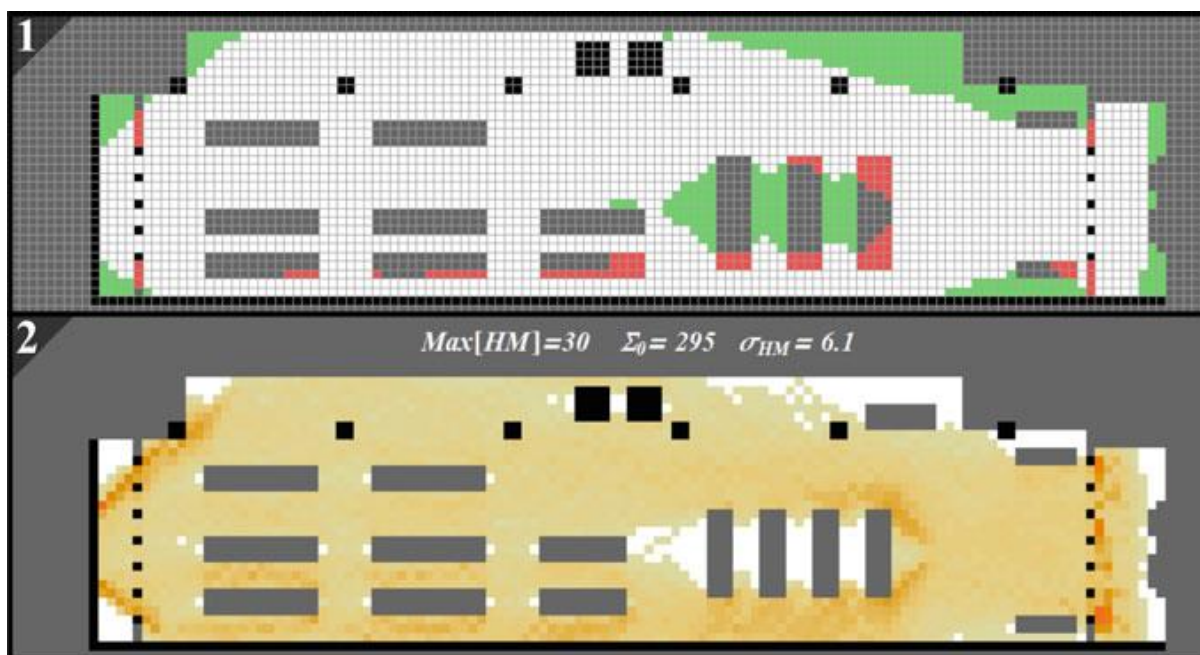
Ponadto artykuł prezentuje potencjalnie zastosowania w obszarze „ekstremalnych habitatów” takich jak stacje kosmiczne, polarne, podwodne itp. oraz kilka wstępnych prototypów.

Jak wspomniano wyżej, badania habilitanta w dyscyplinie informatyka koncertują się na aplikacjach odnoszących się do projektowania, co implikuje ich wielowątkowość. W efekcie, z czasem w sposób naturalny wyodrębniły się jeszcze dwa obszary tworzące oddzielne cykle: numeryczna optymalizacja funkcjonalna planów architektonicznych i urbanistycznych, oraz metody obliczeniowe do pasywnej i aktywnej kontroli elewacji budynku, w szczególności systemów modularnych opartych o automaty komórkowe. Oba tym tematami habilitant poświęcił oddzielne monografie oraz cykle publikacji.

a) Optymalizacja układów architektonicznych i urbanistycznych

1. M. Zawidzki M., 2016, [*Discrete Optimization in Architecture – Architectural & Urban Layout*](#), SpringerBriefs in Architectural Design and Technology, **Springer Singapore**.

Celem tej monografii było zebranie w sposób całościowy i jednolity opracowanych przeze mnie wcześniej nowatorskich koncepcji odnoszących się do układów architektonicznych i urbanistycznych przy zastosowaniu metod inteligencji obliczeniowej w ujęciu optymalizacji dyskretnej. Monografia składa się z dwóch części: poświęconej nowej metodzie opartej o przeszukiwaniu włąb (ang. depth-first search) dyskretnej konfiguracji przy ograniczeniach (ang. constrained satisfaction problem, w skrócie CSP) i na automatycznej analizie jakości układów urbanistycznych, oraz poświęconej symulacji tłumu (ang. crowd simulation) zarówno do analiz „przepływów” ruchu pieszego jak i optymalizacji planów architektonicznych pod kątem płynnego przepływu tłumu i minimalizacji przestrzeni niewykorzystywanych, co ilustruje Fig.35.

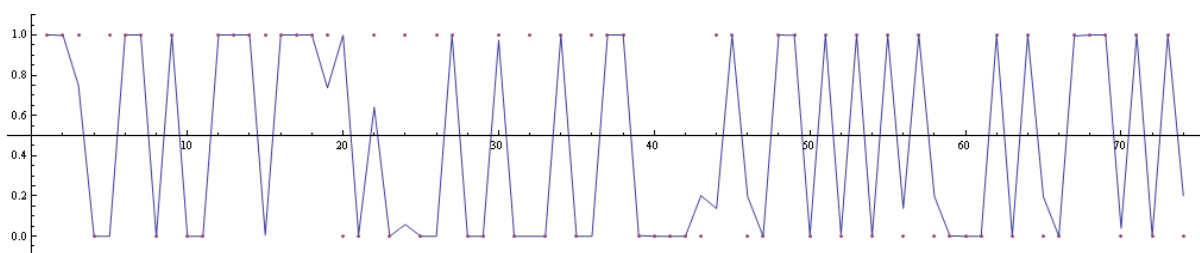


35. Przykład kreatywnego zastosowania automatów komórkowych do kształtowania planu architektonicznego: tłum odwiedzający sklep „roztapia” pewne elementy (zaznaczone na czerwono) w celu wyrównania częstości odwiedzin we wszystkich komórkach planu. W miejscach nieodwiedzanych (zaznaczonych na zielono) wstawiono dodatkowe regały z towarem.

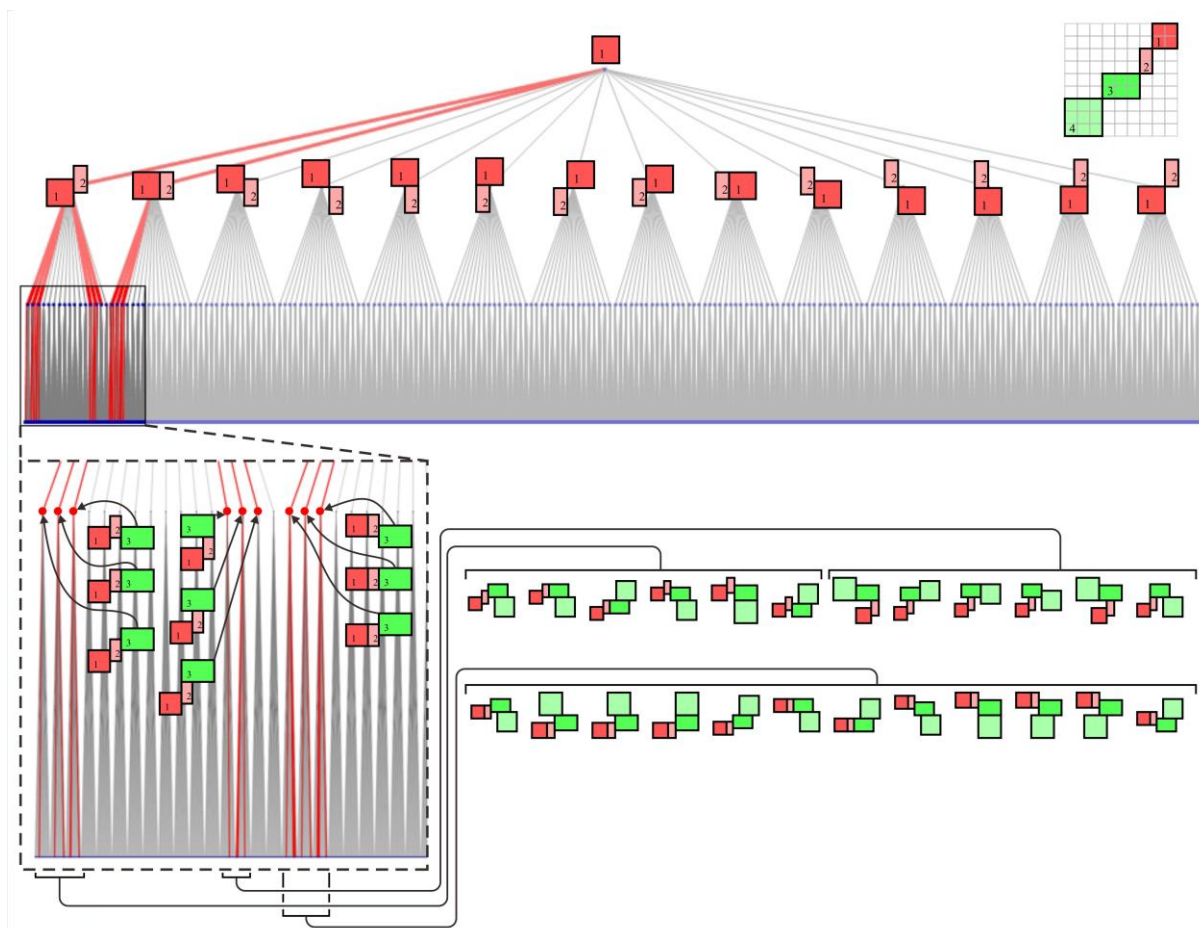
2. M. Zawidzki, K. Tateyama, I. Nishikawa, 2011, [The Constraints Satisfaction Problem approach in the design of an architectural functional layout](#), **ENGINEERING OPTIMIZATION**, 43(9), pp. 943 – 966.
IFs :1.29, **Q2** (Engineering, Multidisciplinary) **Q2** (Operations Research & Management Science), **Lista A MNiSW** – 30 pkt.

Praca ta wprowadza nową koncepcję optymalizacji układów architektonicznych w oparciu o „grubą” siatkę kwadratową (o wymiarach 1-1.5 m), opisuje stworzenie algorytmu opartego o grafową metodę przeszukiwania wgłąb (ang. depth-first search) i jego implementację w środowisku *Mathematica*TM. Fig. 36 ilustruje drzewo wszystkich możliwych i poprawnych konfiguracji 4 pomieszczeń.

Ponadto, artykuł wprowadza zastosowanie sztucznej sieci neuronowej do wstępnej oceny jakościowej układów pomieszczeń w budynku pod kątem możliwości dostępu do wszystkich pomieszczeń poprzez wewnętrzny korytarz. Fig. 37 pokazuje walidację najlepiej wytrenowanej sztucznej sieci neuronowej.

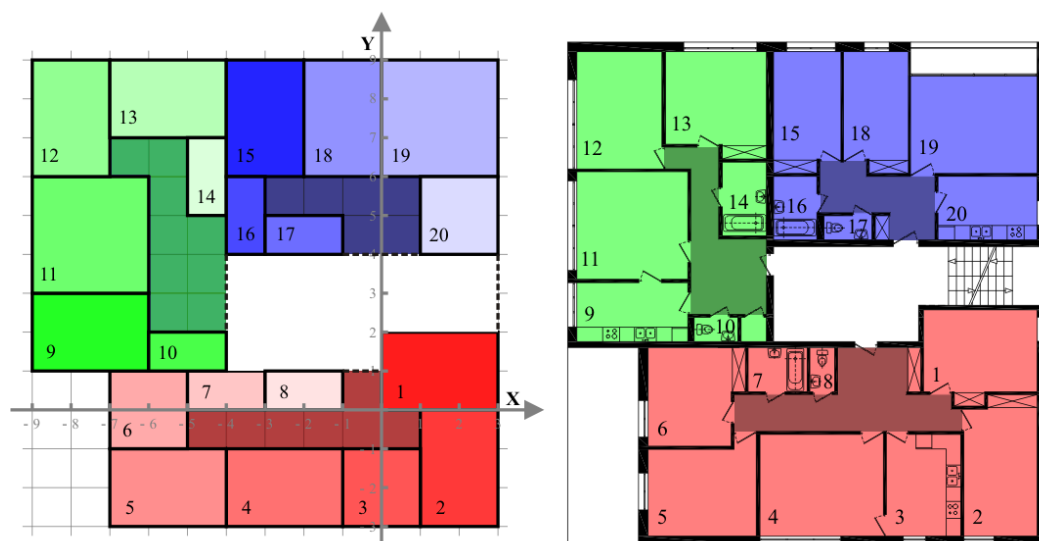


37. Walidacja sztucznej sieci neuronowej wstępnie klasyfikującej jakość układów funkcjonalnych w budynku. Punkty pokazują poprawne wyniki, linia łamana – klasyfikację przez sieć neuronową. Błędy wystąpiły w 8 na 74 próbki (89% poprawność).



36. Drzewo przeszukiwania dla sekwencji pomieszczeń pokazanych w prawym górnym rogu ma 4928 „liści”. Na czerwono pokazano rozwiązania dopuszczalne.

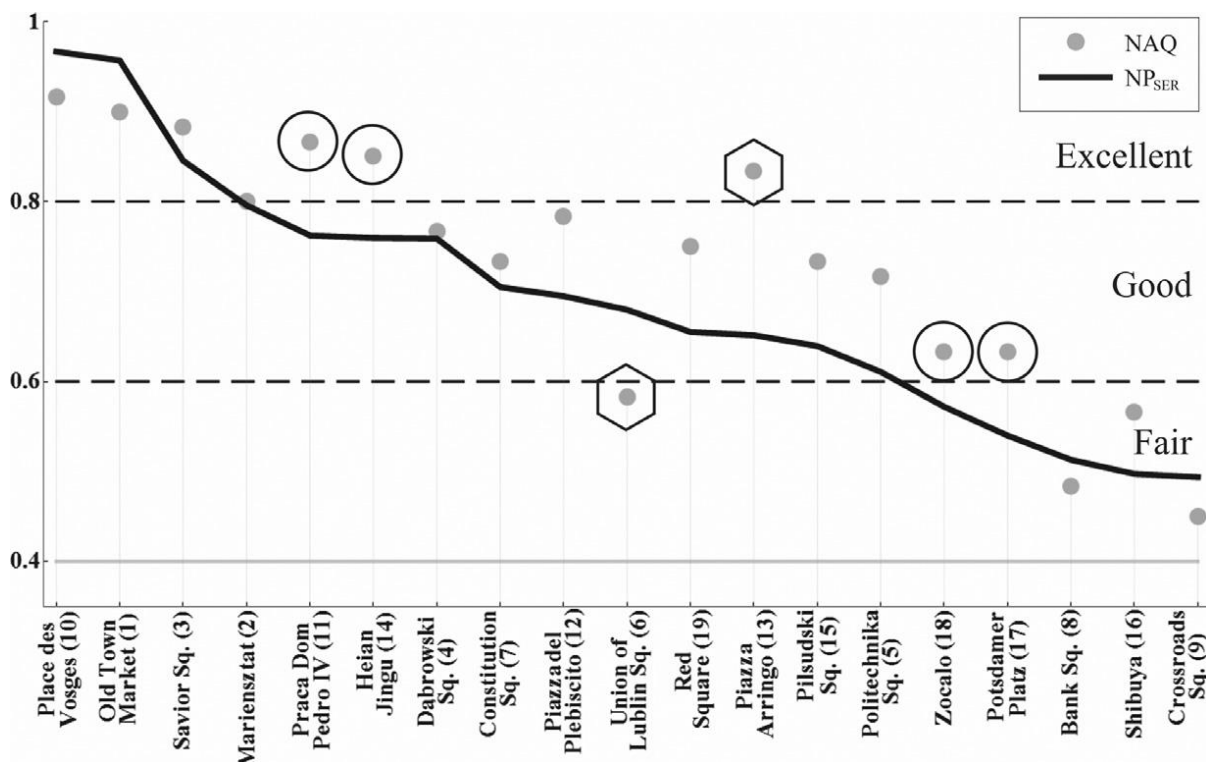
Artykuł wprowadza wielokryterialną funkcję oceniającą ilościowo plany architektoniczne, która ocenia równocześnie: ogólną złożoność planu, wielkość powierzchni korytarzy i preferowane położenie danych pomieszczeń. Przykłady wyjaśniające są dość elementarne. Artykuł wieńczy realistyczne studium przypadku budynku wielorodzinnego z trzema mieszkaniami (łącznie 20 pomieszczeń). Fig. 38 pokazuje najlepsze rozwiązanie.



38. Po lewej: najlepsze rozwiązanie wygenerowane przez proponowany algorytm optymalizujące rzut budynku na „grubej” siatce. Po prawej: plan architektoniczny opracowany na jego podstawie.

3. M. Zawidzki, 2016, [Automated Geometrical Evaluation of a Plaza \(Town Square\)](#), **ADVANCES IN ENGINEERING SOFTWARE**, 96, pp. 58–69, 2016. **IF5:2.54**, **Q1** (Computer Science, Interdisciplinary Applications) **Q1** (Computer Science, Software Engineering) **Q1** (Engineering, Multidisciplinary), **Lista A MNiSW**: 30 pkt. (**dblp**)

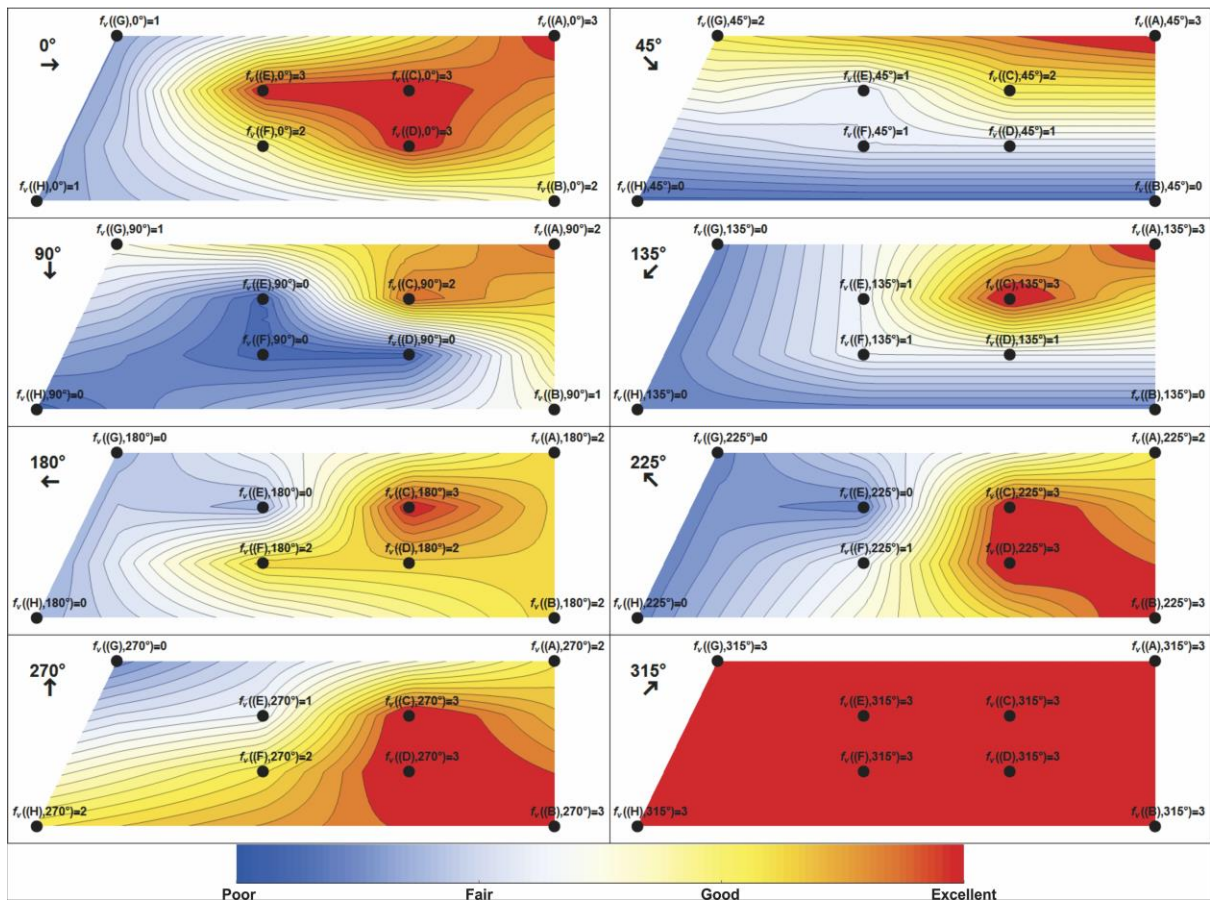
Praca ta prezentuje nowatorskie narzędzie wspomagające projektowanie urbanistyczne oparte na nowym podejściu do zagadnienia oceny jakości placów urbanistycznych. Algorytm opiera się na trzech znormalizowanych własnościach danego placu będących pochodnymi informacji wynikających wyłącznie z jego planu: „małość” (ang. smallness), zwartość, „domknięcie” (ang. enclosure) i regularność. 19 placów poddano ocenie algorytmicznej a wyniki porównano z subiektywną oceną 20-stu respondentów. Wykazano wysoką korelację między proponowaną tą oceną i subiektywną oceną respondentów. Fig. 39 porównuje algorytmiczną ocenę 19 placów z uśrednioną subiektywną oceną respondentów.



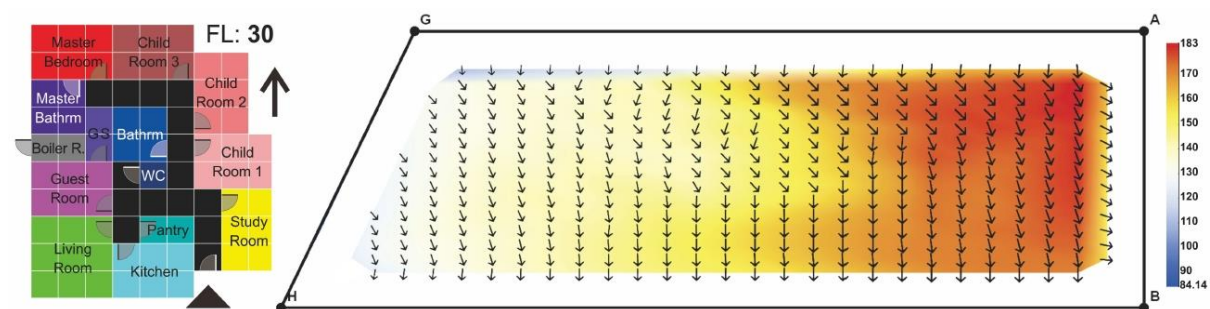
39. 19 placów miejskich od najlepszego do najgorszego. Grubą linią zaznaczono wartości generowane przez algorytm, szare punkty oznaczają uśrednioną subiektywną oceną 20-stu respondentów.

W ostatnim czasie habilitant pracuje nad kolejnymi algorytmicznymi implementacjami zwykle pomijanych (a w swej istocie bardzo ważnych) aspektów przy optymalizacjach architektonicznych w budownictwie jednorodzinnych takich jak komfort akustyczny czy „widokowy”. Fig. 40 pokazuje *heatmapy* jakości widoku na danej działce w ośmiu oktantach.

W tym projekcie optymalizowany jest równocześnie układ pomieszczeń wewnątrz budynku, jego położenie na działce i orientacja. Wyniki optymalizacji ilustruje Fig. 41.



40. Jakość widoku na tej działce zależy od kierunku patrzenia i punktu widokowego. Ostatni wykres (cały czerwony) oznacza, że patrząc w kierunku północno-wschodnim (315°) widoki w każdym punkcie działki są „doskonałe”.



41. Poł lewej: optymalny układ pomieszczeń. Po prawej najlepsze lokalizacje tego budynku wraz z orientacją. Im wyższa wartość funkcji celu tym lepsze rozwiązanie.

Wstępne wyniki zostały zaprezentowane na konferencji: Zawidzki M., Szklarski J., [The Ideal House - multicriterial optimization of a Single Family House](#), ECT 2018: The 10th International Conference on Engineering Computational Technology, 4-6 September 2018, Sitges, Barcelona, Spain. Kompletne wyniki zostały złożone do publikacji (Zawidzki M., Szklarski J., Multicriterial optimization of the floor plan of a single-story family house and its position on the building plot, ADVANCES IN ENGINEERING SOFTWARE, in review. 30 pkt. MNiSW).

b) Kontrola elewacji architektonicznej

1. M. Zawidzki, 2017, [*Discrete Optimization in Architecture – Building Envelope*](#), SpringerBriefs in Architectural Design and Technology, **Springer** Singapore. Monografia indeksowana przez **SCOPUS**.

Celem tej monografii było zebranie w sposób całościowy i jednolity opracowanych przeze mnie wcześniej koncepcji odnoszących się do elewacji w ujęciu optymalizacji dyskretnej ukierunkowanej na dziedzinę architektury i budownictwa. Monografia podzielona jest na dwie części: poświęconej problematyce Elewacji Inteligentnej oraz poświęconej modułowym systemom do kontroli stopnia zaciemnienia elewacji budynków przy zachowaniu ich architektonicznie atrakcyjnej formy (w szczególności opartej o automaty komórkowe). Fig. 42. pokazuje przykład zaciemniania elewacji budynku przez modułarny system oparty o automat komórkowy na siatce kwadratowej.

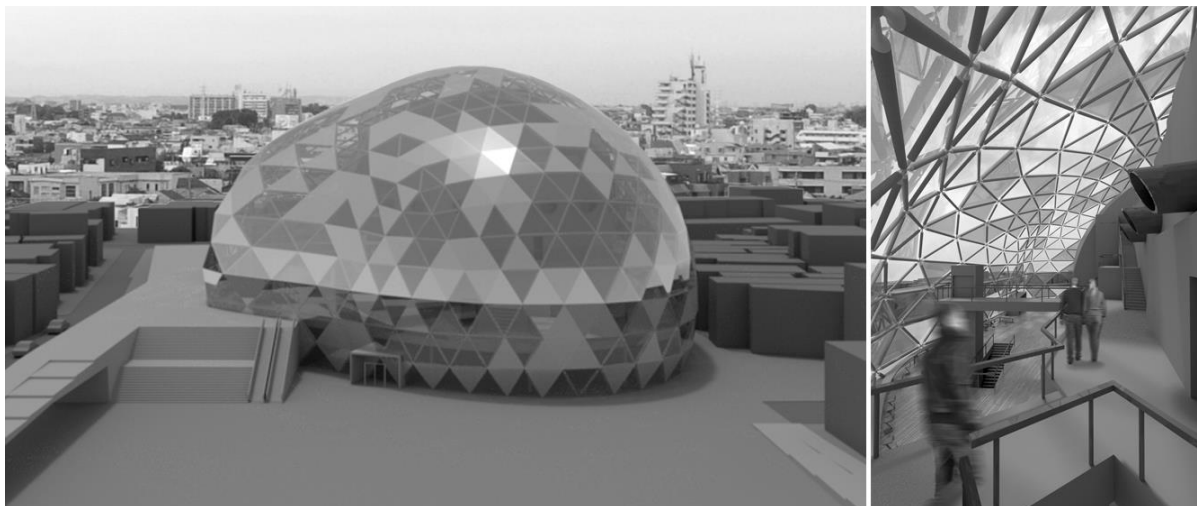


42. Wizualizacja systemu zaciemniania elewacji przez modułarny system oparty o jednowymiarowy automat komórkowy zakresu-2 o kodzie {3818817080, 2, 2} na siatce kwadratowej.

2. M. Zawidzki, 2014, [*Cellular automaton shading for building envelopes*](#), in P. Rosin, A. Adamatzky, and X. Sun, (eds.), **Cellular Automata in Image Processing and Geometry**, pp. 205-231, Springer-Verlag. Series: **Emergence, Complexity and Computation**. Chapter

Celem tego rozdziału było podsumowanie badań habilitanta w dziedzinie modułowych systemów kontroli przezierności (zaciemnienia) elewacji opartych o automaty komórkowe. Przedstawiono analogie między elewacją budynku a biologiczną skórą, funkcje do ilościowej oceny wzorów generowanych przez automaty komórkowe w celu ich optymalizacji: *Grayness Function*, *Grayness Function Error* oraz *Grayness Monotonicity and Pattern Distribution Error*. Ponadto, przedstawiono analizę szeregu typów automatów komórkowych przy użyciu w/w funkcji: elementarnych, połówkowych (ang. half-distance), wyższego rzędu

(ang. higher order). Obok siatki kwadratowej rozważane są również siatki trójkątne i sześciokątne; dwuwymiarowe automaty komórkowe; trójkątne automaty komórkowe na siatce regularnej oraz nieregularnej (Fig. 43). Przedstawiono symulację modularnego systemu zaciemniania na wypukłej nieregularnej ztriangulowanej powierzchni elewacji przy użyciu trzech rodzajów dwukolorowych dwuwymiarowych trójkątnych automatów komórkowych.



43. Symulacja zaciemniania elewacji budynku o nieregularnej formie przy pomocy dwukolorowego dwuwymiarowego trójkątnego automatu komórkowego nr 9622.

3. M. Zawidzki, 2015, [*Dynamic shading of a building envelope based on rotating polarized film system controlled by one-dimensional cellular automata in regular tessellations \(triangular, square and hexagonal\)*](#), **ADVANCED ENGINEERING INFORMATICS**, 29(1), pp. 87-100.
Fs:3.712 Q1 (Computer Science, Artificial Intelligence) Q1 (Engineering, Multidisciplinary),
Lista A MNiSW – 40 pkt. (dblp).

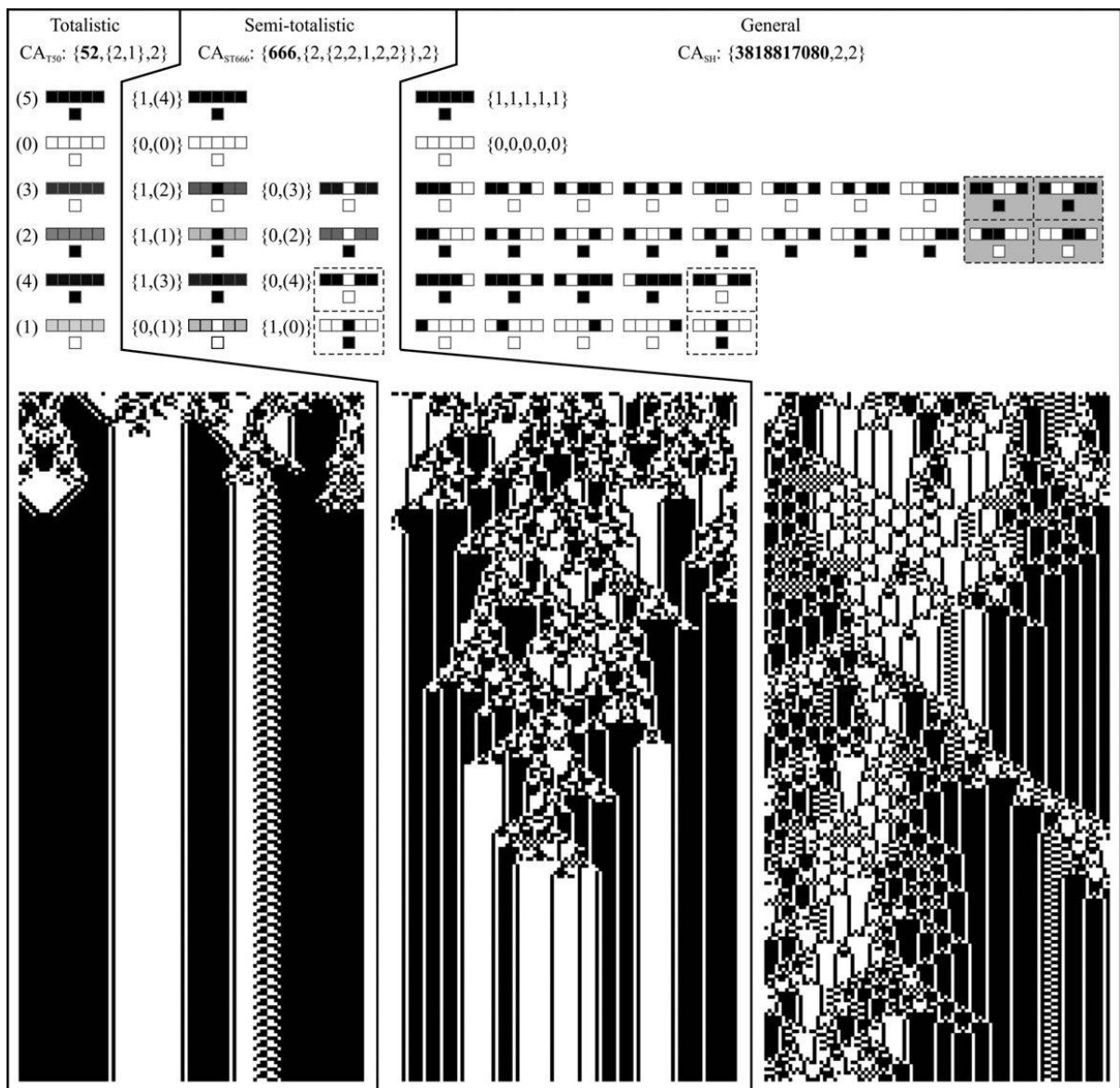
Praca ta dokumentuje badania habilitanta nad koncepcją modularnego systemu elewacyjnego opartego o jednowymiarowe automaty komórkowe na siatkach: trójkątnej, kwadratowej i sześciokątnej. Przeprowadza dyskusję na temat wad i zalet systemów opartych na tych trzech regularnych siatkach, ich występowania w naturze, oraz na temat walorów estetycznych takich systemów – mierzonej przez występowanie *emergencji*, oraz wizualnej atrakcyjności danego rodzaju siatki. Dyskusja jest bogato ilustrowana przykładami. Porównanych pod tym kątem jest szereg typów automatów komórkowych: elementarne, promień-2, trykolorowe, totalistyczne, semi-totalistyczne, połówkowe (ang. half-distance), trykolorowe połówkowe, totalistyczne drugiego rzędu. Przedstawiono szereg symulacji i wizualizacji możliwych rozwiązań. Wykonanie i prezentacja fizycznego urządzenia zbudowanego z 32 identycznych układów elektronicznych emulującego 256 elementarnych automatów komórkowych oraz modularnego panelu zaciemniającego sprzężonego z w/w urządzeniem. Fig. 44. pokazuje przykład zaciemniania elewacji budynku przez modularny system oparty o automat komórkowy na siatce trójkątnej.



44. Wizualizacja systemu zaciemniania elewacji przez modułarny system oparty o jednowymiarowy trzykolorowy trójkątny automat komórkowy 1D3Cr-1/2 kod 226.

4. M. Zawidzki, K. Nishinari, 2013, [*Shading for building facade with two-color one-dimensional range-two cellular automata on a square grid*](#), **JOURNAL OF CELLULAR AUTOMATA**, 8(3-4), pp. 147–163.
F5: 0.504 Q4 (Computer Science, Theory & Methods), **Lista A MNiSW** – 15 pkt. (dblp)

Artykuł ten wprowadza zagadnienie zaciemniania elewacji budynku przez modułarny system oparty o jednowymiarowe dwukolorowe automaty komórkowe na siatce kwadratowej. Poddaje analizie te automaty komórkowe pod kątem ich wizualnej atrakcyjności (reprezentowanej przez emergencję) przy równoczesnej „kontrolowalności” (reprezentowanej przez możliwość względnie kontrolowanej zmiany średniego zaciemnienia całej elewacji poprzez „płynną” modyfikację ustawień warunków początkowych w górnym rzędzie komórek). Ponadto, artykuł analizuje przydatność różnego rodzaju warunków brzegowych (periodyczne, lustrzane, „utwierdzone”) do modułarnego panelu zaciemniającego, oraz rozpatruje dwa rodzaje awarii dla trzech rodzajów automatów komórkowych. Fig. 45 ilustruje trzy automaty komórkowe przydatne w tym systemie.



45. Trzy „spokrewnione” automaty komórkowe o rosnącej złożoności.

5. M. Zawidzki, M. Bator, 2013, [Application of evolutionary algorithm for optimization of the sequence of initial conditions for the cellular automaton-based shading](#),

JOURNAL OF CELLULAR AUTOMATA, 7(5-6), pp. 363–384.

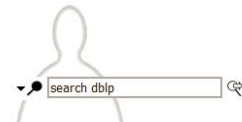
F5: 0.504 Q4 (Computer Science, Theory & Methods), **Lista A** MNiSW – 15 pkt. (dblp)

Artykuł poświęcony jest optymalizacji „sekwencji warunków początkowych” (ang. sequence of initial conditions, w skrócie SIC) w celu jak najbardziej „płynnego” przejścia do stanu zaciemnionego. Opracowano algorytm z nawrotami i znaleziono idealne rozwiązanie dla dla panelu o wymiarach 13×13 komórek. Opracowano algorytm genetyczny i zaimplementowano go w środowisku *Mathematica*TM dla panelu o wymiarach 100×100 komórek, Fig. 46 ilustruje najlepsze rozwiązanie.



46. Sekwencja 100 wzorów dla najlepszego SIC dla panelu 100×100. Poważniejsze zakłócenia występują w krokach: 33, 41 i 63 (zaznaczono ramką).

Lista publikacji w bazie **dblp**



Machi Zawidzki 









> Home > Persons

by year Trier 1







2010 - today

Refine list









2018

- [j14]     Machi Zawidzki, Jacek Szklarski:
Transformations of Arm-Z modular manipulator with Particle Swarm Optimization. Advances in Engineering Software 126: 147-160 (2018)
- [j13]     Machi Zawidzki, Jacek Szklarski:
Effective multi-objective discrete optimization of Truss-Z layouts using a GPU. Appl. Soft Comput. 70: 501-512 (2018)









2016

- [j12]     Machi Zawidzki:
Automated geometrical evaluation of a plaza (town square). Advances in Engineering Software 96: 58-69 (2016)
- [j11]     Machi Zawidzki:
Optimization of Multi-branch Truss-Z based on evolution strategy. Advances in Engineering Software 100: 113-125 (2016)









2015

- [j10]     Machi Zawidzki:
Dynamic shading of a building envelope based on rotating polarized film system controlled by one-dimensional cellular automata in regular tessellations (triangular, square and hexagonal). Advanced Engineering Informatics 29(1): 87-100 (2015)
- [j9]     Machi Zawidzki:
Retrofitting of pedestrian overpass by Truss-Z modular systems using graph-theory approach. Advances in Engineering Software 81: 41-49 (2015)













2014

- [j8]     Machi Zawidzki:
The Influence of Grid Rotation in von Neumann and Moore Neighborhoods on Agent Behavior in Pedestrian Simulation. Complex Systems 23(4) (2014)
- [j7]     Machi Zawidzki, Mohcine Chraïbi, Katsuhiro Nishinari:
Crowd-Z: The user-friendly framework for crowd simulation on an architectural floor plan. Pattern Recognition Letters 44: 88-97 (2014)

2013

- [j6]     Machi Zawidzki, Katsuhiro Nishinari:
Application of evolutionary algorithms for optimum layout of Truss-Z linkage in an environment with obstacles. Advances in Engineering Software 65: 43-59 (2013)
- [j5]     Machi Zawidzki, Katsuhiro Nishinari:
Shading for Building Facade with Two-Color One-Dimension Range-Two Cellular Automata on A Square Grid. J. Cellular Automata 8(3-4): 147-163 (2013)

2012

- [j4]     Machi Zawidzki, Katsuhiro Nishinari:
Modular Truss-Z system for self-supporting skeletal free-form pedestrian networks. Advances in Engineering Software 47(1): 147-159 (2012)
- [j3]     Machi Zawidzki, Marcin Bator:
Application of Evolutionary Algorithm for Optimization of the Sequence of Initial Conditions for the Cellular Automaton-Based Shading. J. Cellular Automata 7(5-6): 363-384 (2012)
- [c2]     Machi Zawidzki, Katsuhiro Nishinari:
Controlling the Opacity of a Building Envelope by a Triangular Two-Color Two-Dimensional Cellular Automaton. ACR1 2012: 194-203

2010



- [j2]     Machi Zawidzki, Ichiro Fujieda:
The Prototyping of a Shading Device Controlled by a Cellular Automaton. Complex Systems 19(2) (2010)
- [c1]     Machi Zawidzki:
A Cellular Automaton Controlled Shading for a Building Facade. ACR1 2010: 365-372

2000 - 2009

2009

- [j1]     Machi Zawidzki:
Implementing Cellular Automata for Dynamically Shading a Building Facade. Complex Systems 18(3) (2009)

Coauthor Index 

  last updated on 2019-01-09 01:41 CET by the dblp team

 data released under the ODC-BY 1.0 license

see also: [Terms of Use](#) | [Privacy Policy](#) | [Imprint](#)

the dblp computer science bibliography is funded by:



M. Zawidzki