



Politechnika Warszawska
Wydział Samochodów i Maszyn Roboczych



Łukasz Tyl

PRACA DYPLOMOWA
magisterska

Wykorzystanie algorytmu ławicowego do optymalizacji konstrukcji

Opiekun naukowy/prowadzący : **prof. dr hab. inż. Mariusz Pyrz**

Warszawa, 6.07.2015 r.

Stworzenie w środowisku Scilab programu wykorzystującego algorytm ławicowy do optymalizacji konstrukcji prętowych

Plan pracy

- ☐ zaprogramowanie algorytmu ławicowego i przetestowanie go na wybranych przykładach
- ☐ stworzenie w programie rozwiązującego zagadnienie równowagi statycznej konstrukcji prętowych metodą elementów skończonych
- ☐ przetestowanie działania programu MES
- ☐ optymalizacja konstrukcji prętowych – przykłady obliczeniowe o różnym stopniu skomplikowania
- ☐ ocena efektywności procedury optymalizacji i wnioski



Opis algorytmu ławicowego



- inspiracja - zachowanie zwierząt w grupach – ptaki poruszające się w kluczu, ławice ryb

- należy do grupy algorytmów niedeterministycznych, tj. generujących rozwiązanie z wykorzystaniem liczb losowych

- Algorytm ławicowy - zwany też optymalizacją rojem cząstek (PSO – Particle Swarm Optimization)

- po raz pierwszy zaprezentowany przez Eberhart'a i Kennedy'ego w 1995 r.

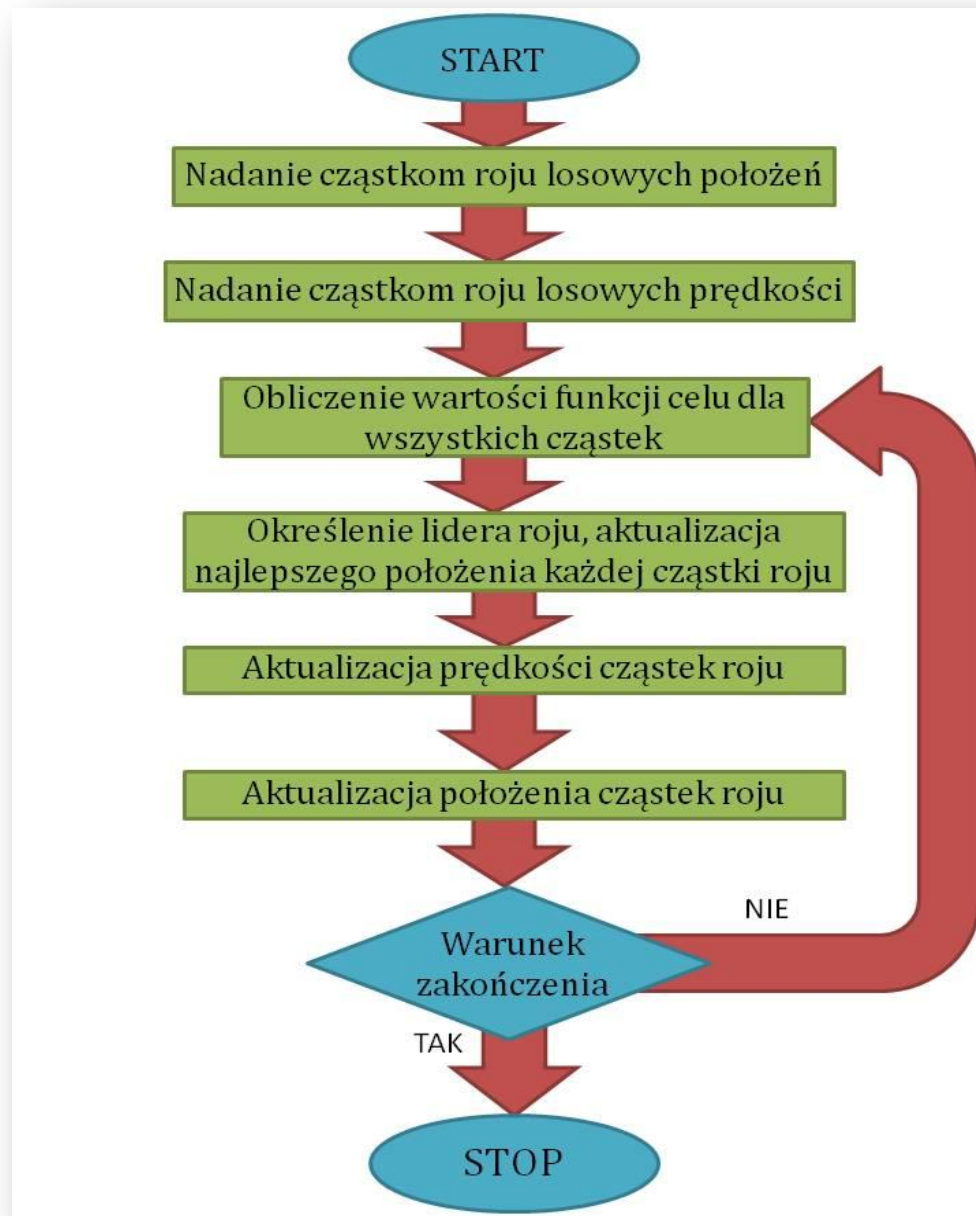


Opis algorytmu ławicowego

Algorytmy ławicowe posługują się specyficzną terminologią, posiadającą jasne odpowiedniki w zagadnieniach klasycznej optymalizacji.

Algorytm ławicowy	Optymalizacja
Cząstka	Pojedyncze rozwiązanie
Współrzędne cząstki	Zmienne projektowe
Rój cząstek	Zbiór rozwiązań
Lider roju	Najlepsze rozwiązanie

Schemat działania algorytmu ławicowego



Opis algorytmu ławicowego

Prędkość i położenie cząstki są obliczane według następujących wzorów:

$$v_{k+1}^i = v_k^i + [c_1 r_{1k}^i (p_k^{li} - x_k^i) + c_2 r_{2k}^i (p_k^g - x_k^i)]$$

$$x_{k+1}^i = x_k^i + v_{k+1}^i$$

x_k^i – wektor położenia i – tej cząstki,

v_k^i – odpowiedni wektor prędkości,

p_k^{li} – najlepsze dotąd znalezione położenie i – tej cząstki,

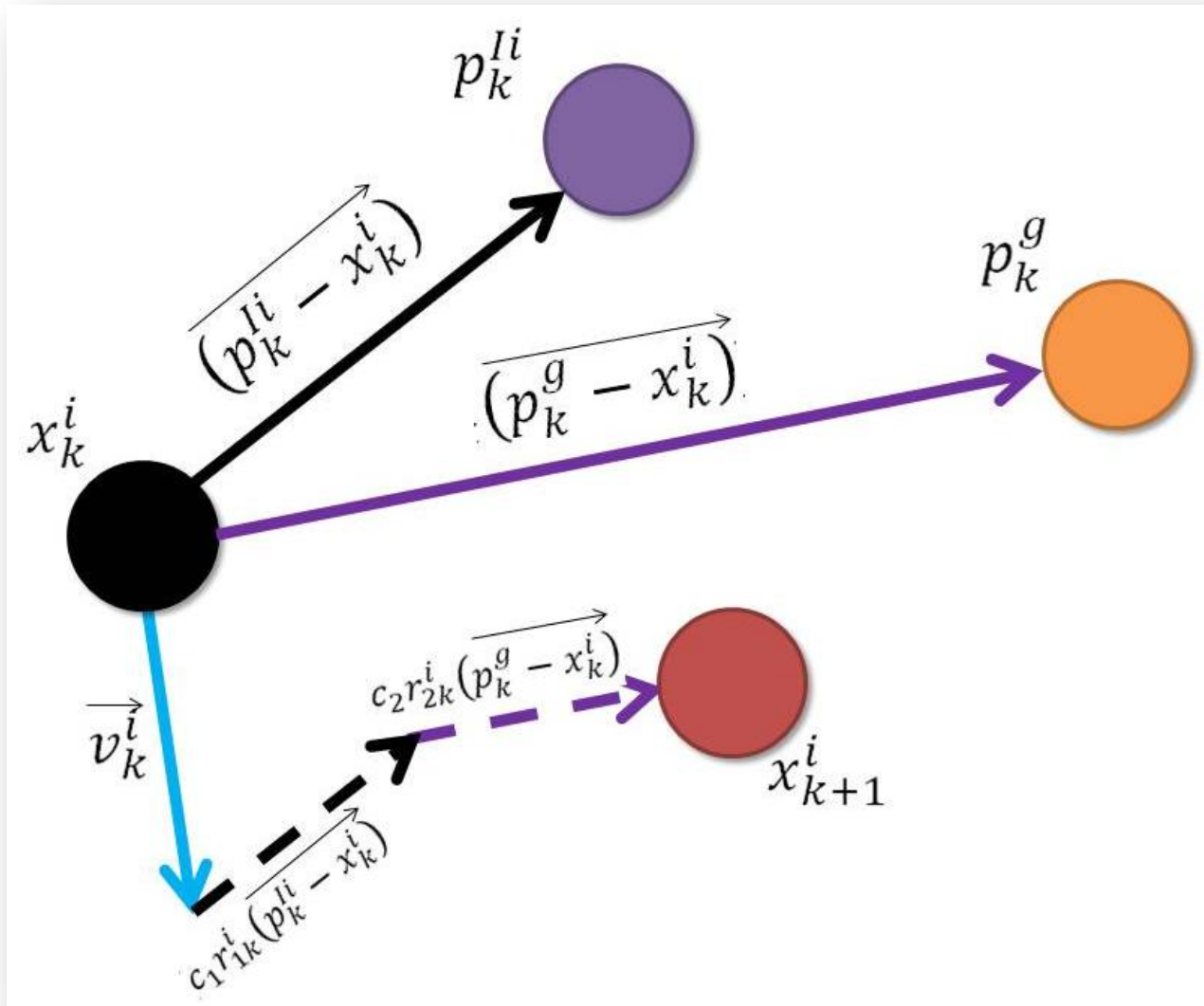
p_k^g – najlepsze dotąd znalezione położenie lidera roju,

c_1, c_2 – ustalone mnożniki wagowe,

r_{1k}^i, r_{2k}^i – liczby losowe z przedziału $(0, 1)$.

Opis algorytmu ławicowego

Interpretacja graficzna aktualizacji ruchu cząstki



Opis algorytmu ławicowego

wersja algorytmu użyta w pracy - **PSO z wagą inercji**

$$v_{k+1}^i = w_k^i v_k^i + [c_1 r_{1k}^i (p_k^{II} - x_k^i) + c_2 r_{2k}^i (p_k^g - x_k^i)]$$

w_k^i – współczynnik wagi inercji, określany na poziomie i – tej cząstki

$$w_k^i = w_{max} - \left(\frac{w_{max} - w_{min}}{it_{max}} \right) \cdot k$$

w_{max} – maksymalna wartość współczynnika wagi inercji

w_{min} – minimalna wartość współczynnika wagi inercji

it_{max} – maksymalna liczba iteracji

i – nr cząstki

k – nr iteracji

Opis algorytmu ławicowego

Pseudokod algorytmu ławicowego

Inicjalizacja początkowej populacji losowej

```
while nie osiągnięto kryterium stopu do
```

```
  for i=1 to rozmiar populacji do
```

```
    if  $f\_celu(x_i) < f\_celu(pbest_i)$  then  
       $pbest_i = x_i$ 
```

```
    if  $f\_celu(x_i) < f\_celu(gbest)$  then  
       $gbest = x_i$ 
```

```
  for d=1 to liczba wymiarów do  
    aktualizacja prędkości  
    aktualizacja położeń
```

W opisie procedur zastosowano następujące oznaczenia:

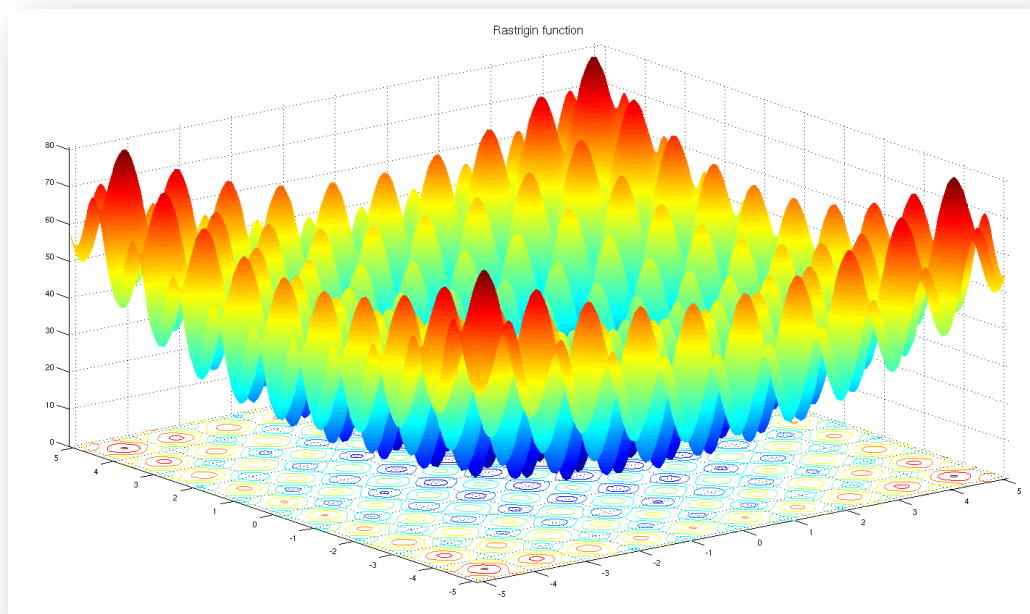
x_i – współrzędne i-tej cząstki (położenie w przestrzeni poszukiwań),

$pbest_i$ – najlepsze dotychczas znalezione położenie i-tej cząstki,

$gbest$ – najlepsze dotychczas znalezione położenie wśród całej populacji cząstek
(lidera roju).

Test algorytmu ławicowego

Do przetestowania działania algorytmu ławicowego, wpływu parametrów na wyniki obliczeń użyto funkcji Rastrigin'a o dwóch zmiennych. Jest to typowy przykład testowy z optymalizacji globalnej, niezwykle trudny do rozwiązania za pomocą klasycznych metod optymalizacji ze względu na wielomodalny (wiele minimów lokalnych) charakter funkcji.



$$f(x) = An + \sum_{i=1}^n [x_i^2 - A \cos(2\pi x_i)]$$

$$A = 10$$

$$x_i \in [-5, 12; 5, 12]$$

$$\text{minimum lokalne} - x = 0, f(x) = 0$$

Test algorytmu ławicowego

Wyniki obliczeń (testu na funkcji Rastrigina) dla przykładowego zestawu parametrów

Parametry	wmax	wmin	itmax	c1	c2	N	D
Wartość	0,9	0,4	100	2	2	100	2

Nr uruchomienia	Wartość funkcji celu	Współrzędne	
	Fbest1	x1	x2
1	0,000545800	0,000417500	0,001605200
2	0,000001300	0,000036500	0,000073600
3	0,994959100	0,000003000	0,994947900
4	0,000000028	0,000007800	0,000008900
5	0,000000000	-0,000000049	-0,000000010
6	0,000000014	-0,000005200	-0,000006600
7	0,007766700	0,005181100	0,003508200
8	0,994959100	0,994958100	-0,000000100
9	0,000000000	-0,000000300	0,000000700
10	0,000000000	0,000001200	0,000000400
Wartość średnia	0,199823204	0,100059965	0,100013819

Program MES w środowisku Scilab

Modelowanie zagadnień statyki konstrukcji prętowych w przestrzeni dwuwymiarowej.

- ☐ Wczytanie danych (geometria i materiał).
- ☐ Budowa macierzy sztywności elementów prętowych.
- ☐ Agregacja globalnej macierzy sztywności konstrukcji.
- ☐ Wprowadzenie warunków brzegowych.
- ☐ Rozwiązanie układu równań.

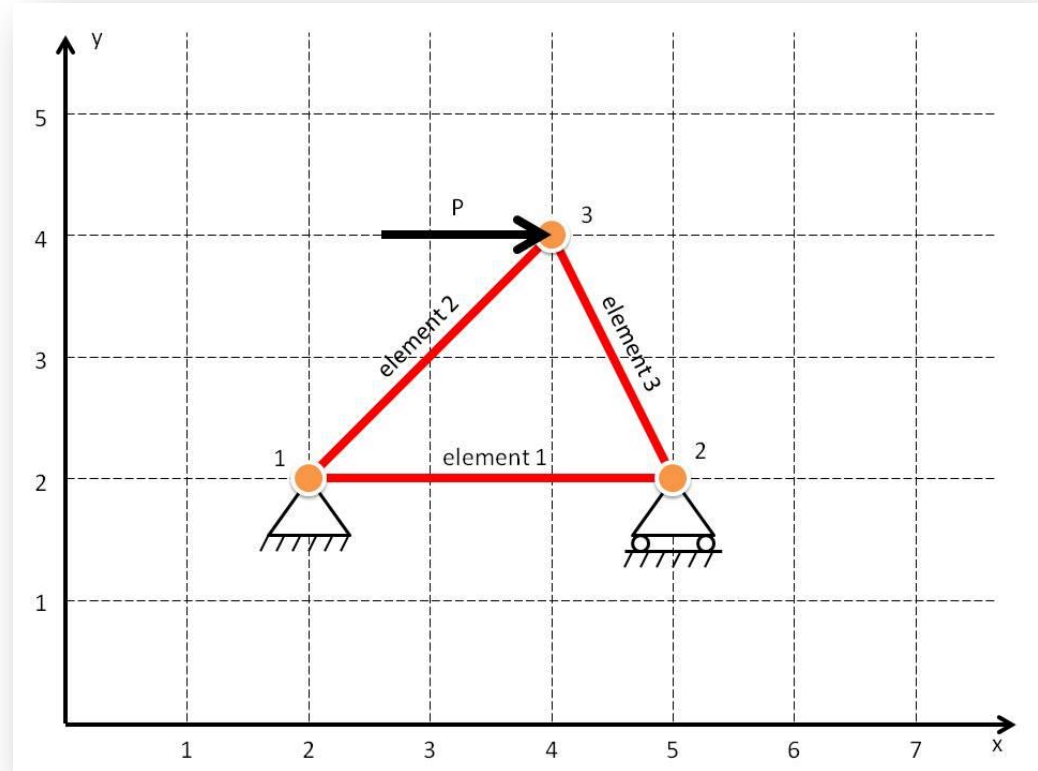
$$K \cdot U = F$$

K – globalna macierz sztywności układu

U – wektor przemieszczeń w węzłach

F – wektor sił działających w węzłach

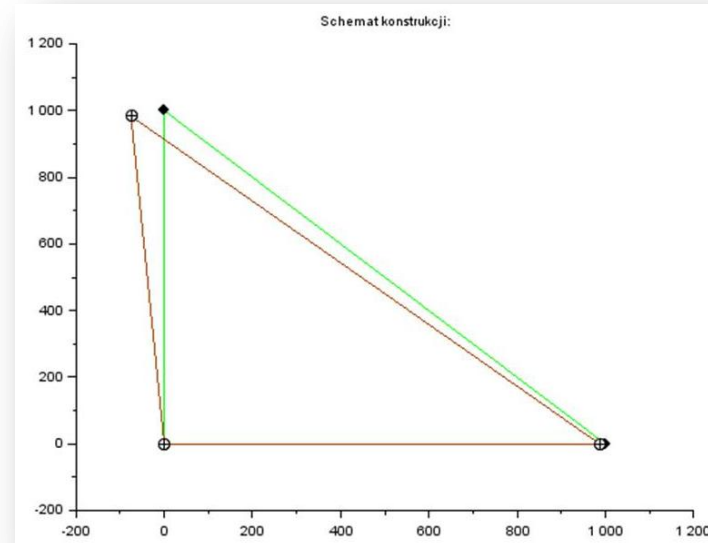
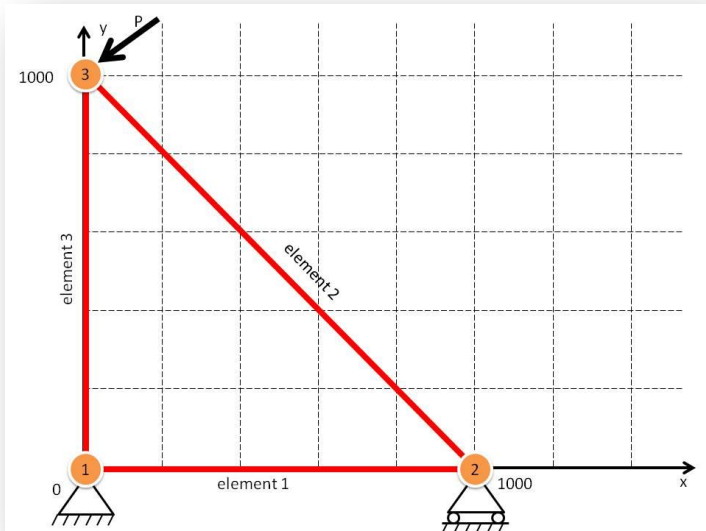
- ☐ Wyznaczanie odkształceń, naprężeń w elementach.



Program MES w środowisku Scilab

Funkcje programu:

- ☐ generowanie prostych rysunków konstrukcji przed i po obciążeniu,
- ☐ obliczanie sumy rzutów sił na osie X i Y oraz sumy momentów względem początku układu współrzędnych,
- ☐ obliczanie naprężeń w elementach skończonych,
- ☐ wyświetlanie komunikatu, gdy naprężenia lub przemieszczenia przekraczają dopuszczalne wartości,
- ☐ obliczanie masy konstrukcji,
- ☐ obliczanie funkcji celu (potrzebnej w procesie optymalizacji).



Program PSO w środowisku Scilab

Kod algorytmu ławicowego (PSO) napisany w środowisku Scilab można wykorzystywać do optymalizacji dowolnej funkcji celu. Docelowo program PSO połączono z modułem MES.

Rozwiązywany problem optymalizacji:	
funkcja celu $F(x) \rightarrow \min$	masa M
ograniczenia: $g(x) \leq 0$	dopuszczalne przemieszczenia $g1$ i naprężenia $g2$
zmienne decyzyjne – wektor x	przekroje prętów

Tryby pracy:

- (a)** zmienne ciągłe (z przedziału $\langle A_{\min}:A_{\max} \rangle$ definiowanego przez użytkownika)
- (b)** zmienne dyskretne (wartości znajdujące się w wektorze `kaTaLogA` definiowanym przez użytkownika)

Sformułowanie zadania optymalizacji

PSO rozwiązuje zadanie optymalizacji bez ograniczeń

- ❑ Konieczność przekształcenia początkowego problemu.
- ❑ Zastosowanie metody funkcji kary.

$$\mathbf{f_{celu}} = (\mathbf{M}/M_{\max}) + (\mathbf{kara} * g1) + (\mathbf{kara2} * g2) ;$$

\mathbf{M}/M_{\max} –stosunek obliczonej masy konstrukcji do masy maksymalnej (tj. masy konstrukcji zbudowanej z elementów o dopuszczalnym maksymalnym przekroju A_{\max}).

$\mathbf{kara} * g1$ –funkcja kary $g1$ dla ograniczeń przemieszczeniowych pomnożona przez odpowiedni współczynnik.

$\mathbf{kara2} * g2$ – funkcja kary $g2$ dla ograniczeń naprężeniowych pomnożona przez odpowiedni współczynnik.

Przykłady optymalizacji konstrukcji prętowych za pomocą algorytmu PSO

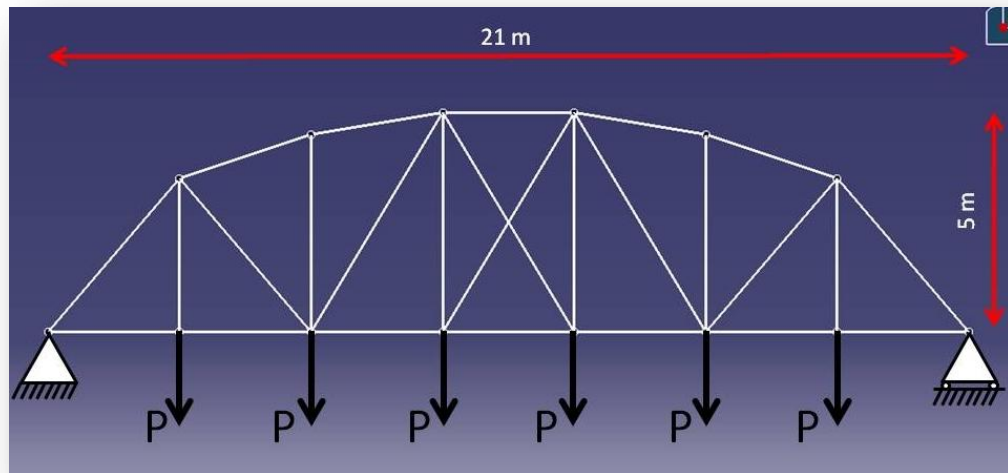
Cel optymalizacji:

minimalizacja masy przy zachowaniu nałożonych na konstrukcje ograniczeń.

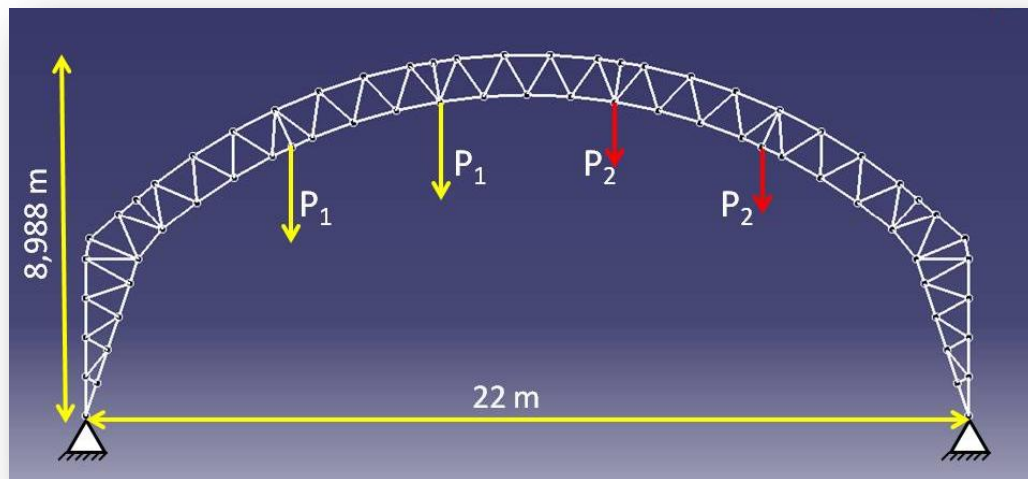
Optymalizacja z wykorzystaniem:

- ☐ zmiennych ciągłych,
- ☐ zmiennych dyskretnych.

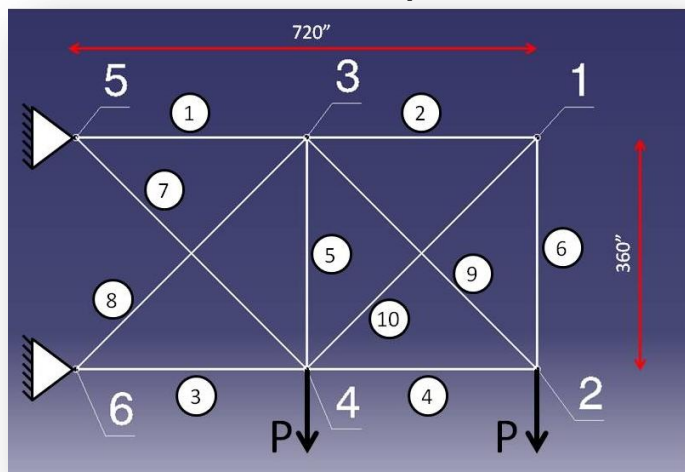
Model płaski przęsła mostu



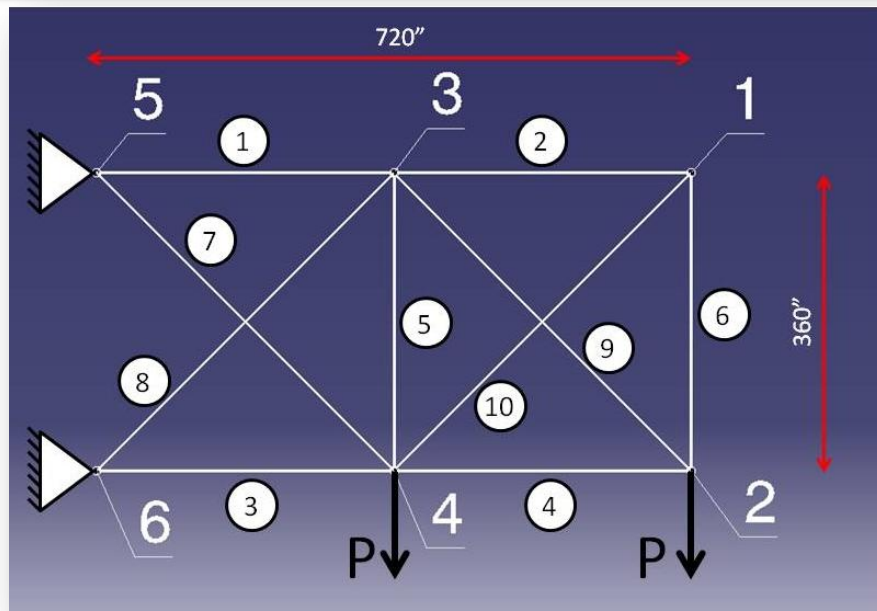
Konstrukcja łukowa



Kratownica płaska



Kratownica płaska



Przykład testowy z literatury
(dane w jednostkach imperialnych):

$$P = 10^5 \text{ lbf}$$

$$E = 10^7 \frac{\text{lbf}}{\text{in}^2} - \text{moduł Young'a}$$

$$\rho = 0,1 \cdot 10^{-3} \frac{\text{lb}}{\text{in}^3} - \text{gęstość}$$

$$\sigma_{dop} = 25000 \frac{\text{lbf}}{\text{in}^2} - \text{naprężenia dopuszczalne}$$

$$u_{dop} = 2 \text{ in} - \text{przemieszczenia dopuszczalne}$$

liczba węzłów: 6; liczba elementów: 10

Zmienne projektowe (przekroje prętów):

-(c) - zmienne ciągłe mogą przyjmować wartość z przedziału $\langle A_{min}:A_{max} \rangle$

$$A_{min} = 0,1 \text{ in}^2$$

$$A_{max} = 36 \text{ in}^2$$

-(d) zmienne dyskretne mogą przyjmować tylko wartości znajdujące się w wektorze **katalogA** (wartości w calach).

katalogA = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36]

Wyniki optymalizacji kratownicy płaskiej

Algorytm ławicowy

	Zestaw nr 1		Zestaw nr 2		Zestaw nr 3		Zestaw nr 4		Bez optymalizacji
	N=100	itmax=200	N=200	itmax=100	N=150	itmax=60	N=200	itmax=100	
ograniczenia	naprężenia, przemieszczenia		naprężenia, przemieszczenia		naprężenia, przemieszczenia		naprężenia		
zmiennie	ciągłe	dyskretne	ciągłe	dyskretne	ciągłe	dyskretne	ciągłe	dyskretne	
masa [lb]	5,112	5,485	5,118	5,291	5,118	5,331	1,848	2,167	8,27

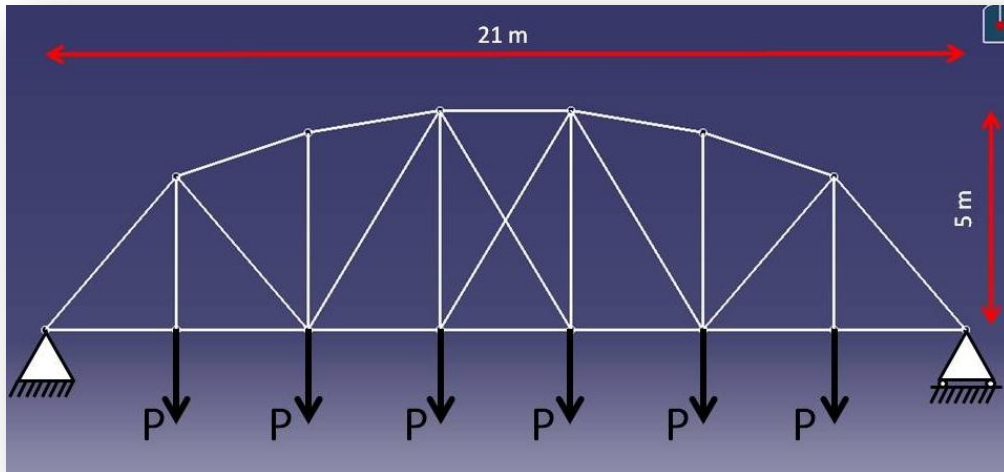
Algorytm ewolucyjny

(wyniki z: *M. Pyrz, Algorytmy ewolucyjne w optymalnym projektowaniu konstrukcji i identyfikacji parametrów materiałowych, 2012*)

	p=100	e=200
ograniczenia	naprężenia, przemieszczenia	
zmiennie	ciągłe	dyskretne
masa [lb]	5,061	5,111

	p=100	e=200
ograniczenia	naprężenia	
zmiennie	ciągłe	dyskretne
masa [lb]	1,593	1,688

Model płaski przęsła mostu



Dane:

$$P = 115000 \text{ N}$$

$$E = 210000 \text{ MPa} - \text{moduł Young'a}$$

$$\rho = 7,86 \cdot 10^{-6} \frac{\text{kg}}{\text{mm}^3} - \text{gęstość}$$

$$\sigma_{dop} = 350 \text{ MPa} - \text{naprężenia dopuszczalne}$$

$$u_{dop} = 30 \text{ mm} - \text{przemieszczenia dopuszczalne}$$

liczba węzłów: 14; liczba elementów: 26

Zmienne projektowe (przekroje prętów):

-(c) zmienne ciągłe mogą przyjmować wartość z przedziału $\langle A_{min}:A_{max} \rangle$

$$A_{min} = 20 \text{ mm}^2$$

$$A_{max} = 4000 \text{ mm}^2$$

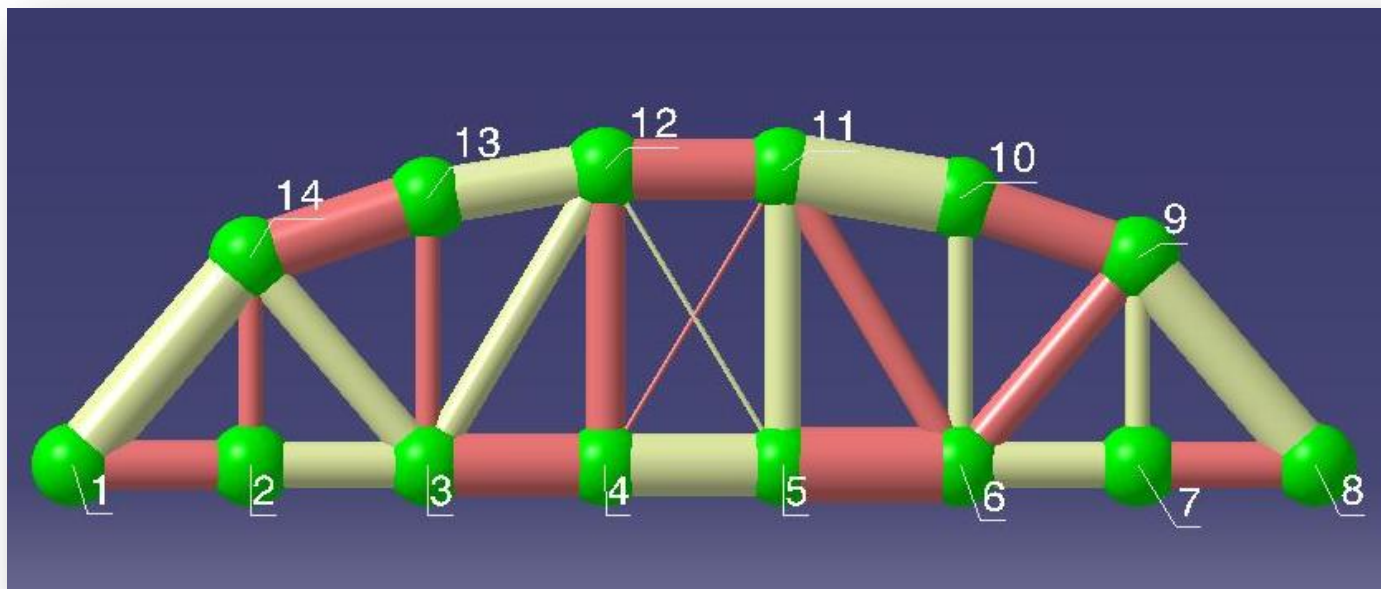
-(d) zmienne dyskretne mogą przyjmować tylko wartości znajdujące się w wektorze **katalogA** (wartości w mm^2).

katalogA = [20, 120, 240, 320, 440, 560, 680, 760, 880, 1000, 1120, 1320, 1560, 1760, 2000, 2240, 2440, 2680, 2880, 3120, 3320, 3560, 3760, 4000]

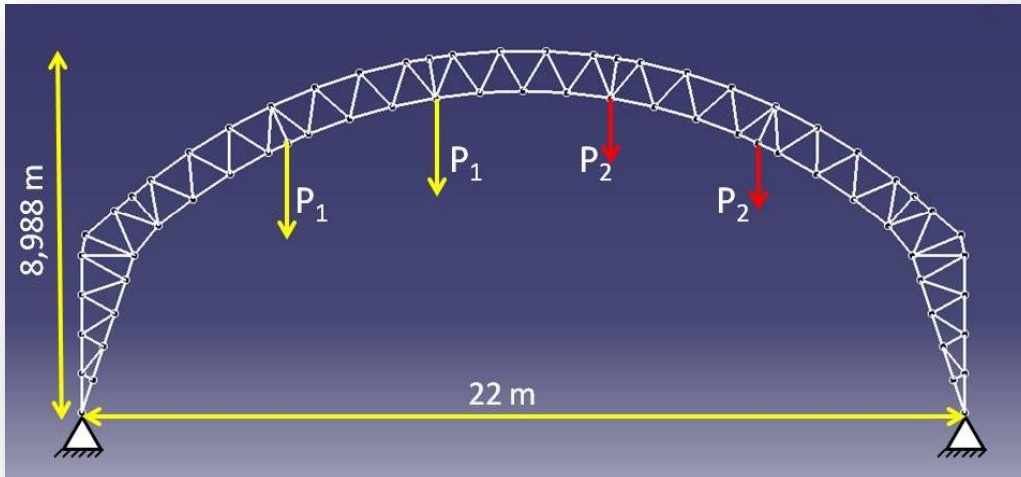
Wyniki optymalizacji modelu przęsła mostu

	Zmienne ciągłe			Zmienne dyskretne			Bez optymalizacji
	Krok 1	Krok 2	Krok 3	Krok 1	Krok 2	Krok 3	
Masa [kg]	1245,2	1232,9	1206	1220,2	1213,3	1193,2	1793,1
Czas obliczeń [s]	82	82	82	220	210	227	-

Graficzne przedstawienie wyników optymalizacji – najlepszy wynik dla zmiennych dyskretnych (grubość pręta proporcjonalna do pola przekroju)



Konstrukcja łukowa



Dane:

$$P_1 = 3000 \text{ N}$$

$$P_2 = 2000 \text{ N}$$

$$E = 70000 \text{ MPa} - \text{moduł Young'a}$$

$$\rho = 2,8 \cdot 10^{-6} \frac{\text{kg}}{\text{mm}^3} - \text{gęstość}$$

$$\sigma_{dop} = 250 \text{ MPa} - \text{naprężenia dopuszczalne}$$

$$u_{dop} = 50 \text{ mm} - \text{przemieszczenia dopuszczalne}$$

liczba węzłów: 67; liczba elementów: 131

Zmienne projektowe (przekroje prętów):

-(c) zmienne ciągłe mogą przyjmować wartość z przedziału $\langle A_{min}:A_{max} \rangle$

$$A_{min} = 10 \text{ mm}^2$$

$$A_{max} = 230 \text{ mm}^2$$

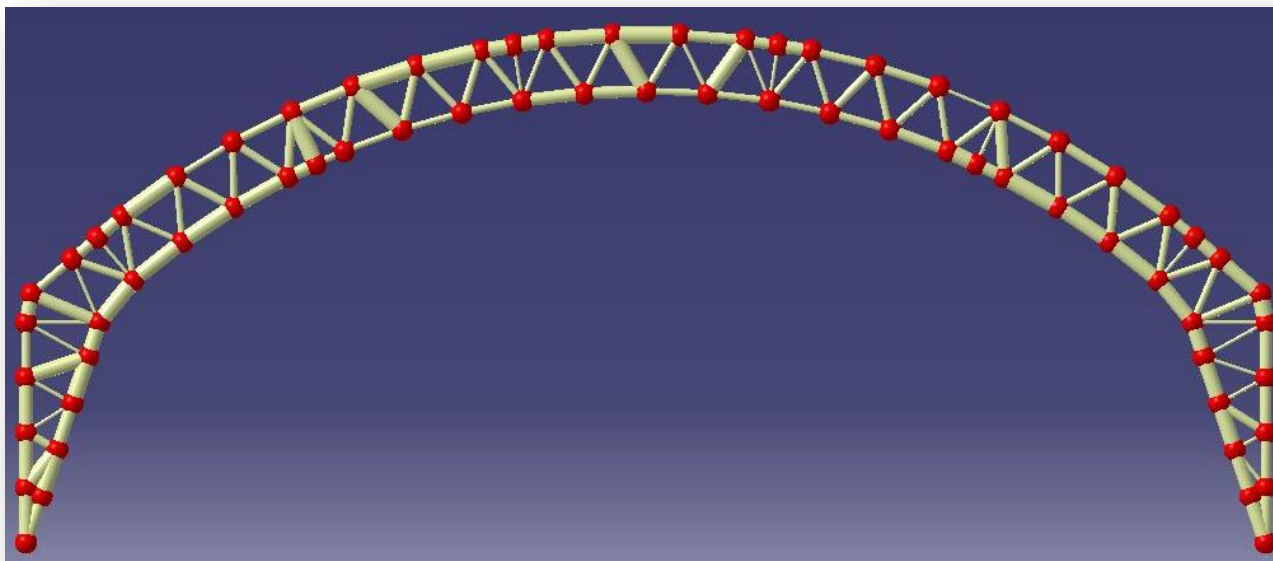
-(d) zmienne dyskretne mogą przyjmować tylko wartości znajdujące się w wektorze **katalogA** (wartości w mm^2).

katalogA = [10, 12, 14, 18, 25, 32, 39, 44, 51, 58, 64, 76, 90, 101, 115, 129, 140, 154, 166, 179, 191, 205, 216, 230]

Wyniki optymalizacji konstrukcji łukowej

	Zmienne ciągłe			Zmienne dyskretne			Bez optymalizacji
	Krok 1	Krok 2	Krok 3	Krok 1	Krok 2	Krok 3	
Masa [kg]	40,56	37,47	34,55	44,51	41,44	32,86	47,4
Czas obliczeń [s]	684	668	668	1401	1634	1607	-

Graficzne przedstawienie wyników optymalizacji – najlepszy wynik dla zmiennych dyskretnych (grubość pręta proporcjonalna do pola przekroju)



- ❑ Zastosowanie optymalizacji w projektowaniu umożliwia osiągnięcie wielu korzyści. W analizowanych przykładach obliczeniowych możliwe było zredukowanie masy o nawet ponad 30% w stosunku do konstrukcji nie optymalizowanej, spełniającej tylko zadane ograniczenia dla jednakowych elementów składowych.
- ❑ Analiza bardziej skomplikowanych przypadków (most, konstrukcja łukowa) przy zwiększającej się ilości węzłów i elementów skończonych znacznie wydłuża się czas obliczeń.
- ❑ Chęć uzyskania zadowalających wyników wymusza stosowanie dużej liczby cząstek oraz liczby iteracji.
- ❑ Dla skomplikowanych przypadków wyniki optymalizacji dla zmiennych dyskretnych są nieznacznie lepsze niż dla zmiennych ciągłych – powód: stosunkowo prosta wersja algorytmu PSO, brak dostatecznej ilości uruchomień procedury optymalizacji.
- ❑ Możliwości rozbudowy programu: konstrukcje trójwymiarowe, zastosowanie innych wersji PSO poprawiających zbieżność algorytmu.

Wykorzystanie algorytmu ławicowego do optymalizacji konstrukcji

Łukasz Tyl

