

## CW 12 Algorytm Innowacyjnego Artylerzysty (AIG)

### 1. Funkcje testowe

- Funkcja Ackley'a

$$f(x) = -20 \exp \left[ -\frac{1}{5} \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right] - \exp \left[ \frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right] + 20 + e$$

gdzie:  $n = 1, 2, \dots$ ; and  $-32.768 \leq x_i \leq 32.768$  for  $i = 1, 2, \dots, n$ . Minimum globalne funkcji  $f_* = 0$  osiągane jest dla  $x_* = (0, 0, \dots, 0)$ .

- Funkcja Rosenbrocka:

$$f(x) = \sum_{i=1}^{n-1} \left[ (x_i - 1)^2 + 100(x_{i+1} - x_i^2)^2 \right]$$

gdzie:  $n = 1, 2, \dots$ ; and  $-30.0 \leq x_i \leq 30.0$  for  $i = 1, 2, \dots, n$ . Minimum globalne funkcji  $f_* = 0$  osiągane jest dla  $x_* = (1, 1, \dots, 1)$ .

- Funkcja DeJonga:

$$f(x) = \sum_{i=1}^n x_i^2$$

gdzie:  $n = 1, 2, \dots$ ; oraz  $-100.0 \leq x_i \leq 100.0$  dla  $i = 1, 2, \dots, n$ . Minimum globalne funkcji  $f_* = 0$  osiągane jest dla  $x_* = (0, 0, \dots, 0)$ .

- Funkcja Griewanka

$$f(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (13)$$

gdzie:  $n = 1, 2, \dots$ ; oraz  $-600.0 \leq x_i \leq 600.0$  dla  $i = 1, 2, \dots, n$ . Minimum globalne funkcji  $f_* = 0$  osiągane jest dla  $x_* = (0, 0, \dots, 0)$ .

- Funkcja Rastrigina :

$$f(x) = 10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)] \quad \text{gdzie: } n = 1, 2, \dots; \text{ oraz } -5.12 \leq x_i \leq 5.12 \text{ for } i = 1, 2, \dots, n.$$

Minimum globalne funkcji  $f_* = 0$  osiągane jest dla  $x_* = (0, 0, \dots, 0)$ .

- Funkcja Schwefela:

$$f(x) = -\sum_{i=1}^n x_i \sin(\sqrt{|x_i|})$$

gdzie:  $n = 1, 2, \dots$ ; oraz  $-500 \leq x_i \leq 500$  dla  $i = 1, 2, \dots, n$ . Minimum globalne funkcji  $f_* \approx -418.9829 \cdot n$  osiągane jest dla  $x_* = (420.9687, 420.9687, \dots, 420.9687)$ . W celu zapewnienia dodatniości funkcji dla

wszystkich numerów wymiarów oraz zmiennych, następująca modyfikacji funkcji powinna podlegać zastosowaniu.

$$f(x) = -\frac{1}{n} \sum_{i=1}^n x_i \sin(\sqrt{|x_i|}) + 500$$

Minimum globalne funkcji  $f_* \approx 81.0171$  występuje w tym samym punkcie, co dla funkcji oryginalnej, tj.  $x_* = (420.9687, 420.9687, \dots, 420.9687)$ .

Celem optymalizacji każdej funkcji jest jej minimalizacja. Algorytmy były jednak budowane tak, aby maksymalizować funkcję przystosowania. Konieczne było więc przejście między funkcją celu a przystosowania, dokonywane docelowo za pomocą:

$$F_{fitness} = \frac{1}{F_{obj} + 1}$$

## 2. Przebieg ćwiczenia

Na bazie przedstawionego w kolejnych punktach opisie algorytmu AIG zespół tworzy w dowolnie wybranym języku program umożliwiający wykonanie optymalizacji funkcji testowych i wykonanie programu ćwiczenia analogicznie jak miało to miejsce dla CW 8-11.

Źródło algorytmu: Paweł Pijarski, Optymalizacja heurystyczna w ocenie warunków pracy i planowaniu rozwoju systemu elektroenergetycznego, Lublin 2019

Program ćwiczenia – przypomnienie z ćw 8-11 na przykładzie optymalizacji algorytmem PSO

- Oprócz funkcji domyślnej zespół wybiera 3 inne funkcje testowe
- W pliku prototype.go należy ustawić odpowiednią dla danej funkcji testowej liczbę zmiennych oraz zakresy zmiennych wejściowych
- Wybór kombinacji parametrów wejściowych w pliku auto\_PSO.bxt, umożliwiających zbadanie wpływu każdej zmiennej na efekty symulacji. Kombinacje pozostają takie same dla każdej funkcji testowej.
- Porównanie wyników na podstawie plików report\_PSO\_auto.txt, w tym ocena jakości rozwiązania znalezionejgo oraz analiza zbieżności

### **3. Nowa autorska metoda optymalizacji heurystycznej – Algorytm Innowacyjnego Artylerzysty (ang. *Algorithm of the Innovative Gunner – AIG*)**

#### **3.1 Uwagi wstępne**

W niniejszym rozdziale przedstawiono nową metodę optymalizacji heurystycznej, dla której inspiracją jest taki dobór parametrów działa artylerystycznego, by kolejny wykonany strzał precyzyjnie osiągnął cel. Zastosowana korekta parametrów ustawienia działa jest jednak inna niż określa to klasyczna teoria artylerii. Stąd w nazwie metody sformułowanie „innowacyjny...” odnoszące się do artylerzysty. Innowacyjność algorytmu wynika z faktu zastosowania, w każdym kroku procesu iteracyjnego, korekty poprzedniego rozwiązania za pomocą specjalnie dobranych mnożników. Stanowi to zasadniczą różnicę w stosunku do większości metod, w których w kolejnym kroku element korygujący jest dodawany do poprzedniego rozwiązania. Zaprezentowany algorytm (ang. AIG) charakteryzuje się wysoką skutecznością i szybkością w rozwiązywaniu różnych zadań optymalizacyjnych, niemających nic wspólnego z balistyką i artylerią. Jest on przy tym niezwykle prosty w opisie i implementacji. Algorytm został przetestowany na znanych, testowych zagadnieniach optymalizacyjnych. Uzyskane wyniki są zauważalnie lepsze od osiągniętych innymi metodami i udowadniają praktyczną przydatność algorytmu AIG (rozdział ten zawiera rozszerzony opis algorytmu opublikowanego w czasopiśmie *Engineering Optimization* [166]).

Poszukiwanie nowych metod inteligencji obliczeniowej pozwala zwiększać skuteczność rozwiązywania szczególnie trudnych i nietypowych zadań obliczeniowych i problemów, dla których zastosowanie metod klasycznych [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19, 20] zawodzi. Proponowana metoda pośrednio bazuje na zjawiskach przyrodniczych, takich jak grawitacja, opór powietrza czy prędkość i kierunek wiatru. Bezpośrednio natomiast dotyczy skuteczności namierzania celu i wstępnego korygowania lotu pocisku (pocisków) za pomocą odpowiedniego ustawienia broni (działu).

**Konfrontując tę metodę z algorytmami już istniejącymi należy podkreślić, że prezentowane podejście nie było jeszcze dotychczas rozważane i analizowane. W związku z tym autor uważa je za własne oryginalne osiągnięcie naukowe.**

Biorąc powyższe pod uwagę można postawić następującą tezę:

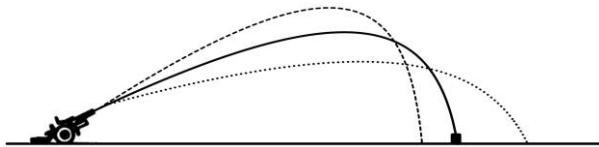
*Algorytmy heurystyczne, w tym autorski algorytm AIG, mogą być efektywnie wykorzystywane do rozwiązywania zadań optymalizacji w elektroenergetyce, ponieważ pozwalają skutecznie minimalizować funkcje celu z dyskretnymi i ciągłymi zmennymi decyzyjnymi przy różnych ograniczeniach funkcyjnych (liniowych i nieliniowych oraz równościowych i nierównościowych).*

Teza zostanie udowodniona w rozdziałach 3–14 poprzez:

- wykonanie testów obliczeniowych nowego algorytmu. Testy będą obejmowały różne funkcje celu z różnych dziedzin naukowych,
- odpowiedni dobór i analizę wybranych zagadnień optymalizacyjnych w elektroenergetyce,
- porównanie wyników obliczeń z wynikami uzyskanymi za pomocą innych algorytmów, uznawanych za klasykę heurystyki.

### 3.2 Inspiracja

Zadaniem jest strzelić z działa pod takim kątem, aby przy losowych zakłóceniach trafić w cel.



Rys. 3.1 Inspiracja algorytmu AIG

Chociaż jak wspomniano wcześniej, inspiracją dla większości metod heurystycznych jest natura, to w przypadku prezentowanego algorytmu AIG zwanego też „balistycznym” inspiracją był pewien pomysł matematyczny, dla którego z czasem określono interpretację bazującą na teorii rzutu ukośnego będącą podstawą balistyki. Rozpatrując zadanie optymalizacyjne, w którym funkcja celu  $F_c$  jest funkcją wektora zmiennych decyzyjnych

$$[\mathbf{x}] = [x_1, \dots, x_l, \dots, x_n] \quad (3.1)$$

Poszukuje się takich wartości tych zmiennych, które zapewnią minimalną wartość funkcji celu, tj.

$$F_c(\mathbf{x}) = F_c(x_1^s, \dots, x_l^s, \dots, x_n^s) \rightarrow \min \quad (3.2)$$

Nawet wobec niezwykle dużej liczby metod, które zapewniają rozwiązanie tego zadania, można stwierdzić, że proces iteracyjny określania optymalnych wartości składowych wektora  $[\mathbf{x}]$  jest podobny, albowiem najczęściej w kolejnym kroku iteracyjnym wykonywane jest działanie

$$x_l^{(i+1)} = x_l^{(i)} + \Delta x_l^{(i)} \quad (3.3)$$

zaś korekta składowej  $l$ -tej wektora, w  $i$ -tym kroku iteracyjnym, wyznaczana jest jako wartość funkcji

$$\Delta x_l^{(i)} = f_l(\xi) \quad (3.4)$$

Należy zauważyc, że zarówno postać funkcji  $f_l$ , jak i też składowe wektora  $\xi$  to symboliczny zapis i cecha charakterystyczna zastosowanej metody heurystycznej, jednej z wielu wskazanych wyżej.

W oparciu o powyższą zależność, obliczenia wykonywane są np. w metodach:

- roju cząstek (PSO),

$$\mathbf{x}_{k+1}^i = \mathbf{x}_k^i + \mathbf{v}_{k+1}^i \quad (3.5)$$

$$\mathbf{v}_{k+1}^i = w \cdot \mathbf{v}_k^i + c_1 \cdot r_1 \cdot (\mathbf{p}^i - \mathbf{x}_k^i) + c_2 \cdot r_2 \cdot (\mathbf{p}^g - \mathbf{x}_k^i) \quad (3.6)$$

gdzie:

$\mathbf{x}_k^i$  – jest wektorem położenia  $i$ -tej cząstki w  $k$ -tej iteracji,

$\mathbf{v}_k^i$  – jest wektorem prędkości  $i$ -tej cząstki w  $k$ -tej iteracji,

$\mathbf{p}^i$  – jest najlepszym, dotychczas znalezionym położeniem  $i$ -tej cząstki,

$\mathbf{p}^g$  – jest najlepszym położeniem znalezionym przez lidera roju,

$w$  – jest współczynnikiem bezwadności ruchu cząstki,

$c_1, c_2$  – są to ustalone współczynniki przyspieszenia lub uczenia się,

$r_1, r_2$  – są liczbami uzyskiwanymi z generatora liczb losowych o rozkładzie równomiernym w przedziale  $[0,1]$ .

Wzór (3.5) może budzić wątpliwość, ponieważ do położenia dodawana jest prędkość. Należy jednak zauważyc, że aktualizacja położenia każdej cząstki dokonywana jest po jednej iteracji algorytmu, więc jest on prawdziwy dla  $t = 1$  s. Dlatego w literaturze spotyka się taką jego postać.

- algorytmu ekosystemowego,
- algorytmu nietoperzy,
- algorytmu grawitacyjnego,
- algorytmu optymalizacji spiralnej, itp.

W niektórych algorytmach, nowe rozwiązanie znajdowane jest w oparciu o sumę poprzedniego i losowo wyznaczanego kroku (odbywa się to poprzez losowanie nowego punktu w otoczeniu bieżącego), który określany jest według ogólnej zasady:

$$x_l^{(i+1)} = x_l^{(i)} + \omega_l^{(i)} \quad (3.7)$$

gdzie  $\omega_l^{(i)}$  jest krokiem lub liczbą losową z zadanego rozkładu. Według powyższej zasady działają algorytmy, takie jak np.:

- przeszukiwanie kukułczane,
- symulowane wyżarzanie,

- algorytmy świętokrzyskie,
- algorytmy mrówkowe,
- przeszukiwanie z tabu,
- algorytmy pszczele, itp.

Inne algorytmy znajdują kolejne rozwiązanie poprzez odpowiednią modyfikację poprzednich, np.:

- algorytmy genetyczne, nowe punkty powstają poprzez reprodukcję, mutacje, krzyżowanie poprzednich,
- sztuczne systemy immunologiczne,
- sztuczne sieci neuronowe, itp.

**Pomysł na algorytm nazwany AIG** polega na tym, by tak popularną i intuicyjną zależność (3.3, 3.7) zastąpić zależnością

$$x_l^{(i+1)} = x_l^{(i)} \cdot g_l(\xi) \quad (3.8)$$

lub nawet

$$x_l^{(i+1)} = x_l^{(i)} \cdot g_{l1}(\xi_1) \cdot \dots \cdot g_{lp}(\xi_p) \quad (3.9)$$

Oznacza to, że w kolejnych krokach procesu iteracyjnego, składowe wektora decyzyjnego podlegają modyfikacjom o charakterze „multiplikatywnym”, w przeciwieństwie do modyfikacji „addytywnych” typu (3.3), które wydają się intuicyjne, niezależnie od specyfiki zastosowanego algorytmu. Jak pokazano w dalszej części monografii, można wskazać bardzo proste funkcje  $g_{l1}, \dots, g_{lp}$  oraz ich argumenty  $\xi_1, \dots, \xi_p$ , które zapewniają efektywne rozwiązanie wielu zadań optymalizacyjnych, traktowanych jako modele banchmarkowe.

### 3.3 Opis „algorytmu innowacyjnego artylerzysty”

#### 3.3.1 Rzut ukośny – fizyczne podstawy balistyki

Zgodnie z elementarnymi pojęciami z fizyki, rzut ukośny to ruch ciała w jednorodnym skierowanym pionowo, polu grawitacyjnym z prędkością początkową o kierunku ukośnym do kierunku pola. Ciało, któremu nadano prędkość  $v_0$  i wystrzelono pod katem  $\delta$  do poziomu (płaszczyzna pozioma jest prostopadła do kierunku pola grawitacyjnego), porusza się po paraboli o równaniu (w prostokątnym układzie współrzędnych  $(d, h)$ ) widocznym poniżej –  $g$  oznacza przyspieszenie ziemskie

$$h = \tan \delta \cdot d - \frac{g \cdot d^2}{2 \cdot v_0^2 \cdot \cos^2 \delta} \quad (3.10)$$

Trafienie w cel oznacza, że dla zadanej odległości poziomej  $d$  i prędkości  $v_0$ , wysokość  $h$  jest równa zeru (cel znajduje się na płaszczyźnie poziomej).

Wymagana wartość kata  $\delta$  może być wyznaczona z zależności

$$\delta = \frac{1}{2} \cdot \arcsin \frac{d \cdot g}{v_0} \quad (3.11)$$

Zadanie komplikuje się, jeśli brany jest pod uwagę opór powietrza. Dla najprostszego modelu (opór powietrza proporcjonalny do prędkości ciała) postać wzoru (3.10) ulega istotnej zmianie, albowiem

$$h = \left( \tan \delta + \frac{g}{k \cdot v_0 \cdot \cos \delta} \right) \cdot d + \frac{g}{k^2} \cdot \ln \left( 1 - \frac{d \cdot k}{v_0 \cdot \cos \delta} \right) \quad (3.12)$$

przy czym wielkość  $k$  jest zdefiniowana jako współczynnik oporu (jednostka 1/s), którego wartość zależy od masy ciała i oporu powietrza. W tym przypadku, aby wyznaczyć kąt strzału, zapewniający trafienie do celu, należy, przyjmując  $h = 0$ , rozwiązać równanie nieliniowe, powstałe z wyrażenia (3.12). Ponieważ historia balistyki sięga czasów Galileusza i Newtona [209], opracowano szereg metod, analitycznych i empirycznych pozwalających zapewnić celność salwy artyleryjskiej. Należy jednak zauważyć, że krzywa lotu rzeczywistego pocisku ma postać

$$h = \left( \tan \delta + \frac{g}{k \cdot v_0 \cdot \cos \delta} \right) \cdot d + \frac{g}{k^2} \cdot \ln \left( 1 - \frac{d \cdot k}{v_0 \cdot \cos \delta} \right) + f_h(\gamma) \quad (3.13)$$

gdzie  $f_h(\gamma)$  jest funkcją odpowiadającą oddziaływaniu czynników zakłócających, takich jak zmieniający się opór powietrza, temperatura, wiatr itp. W takim przypadku kąt  $\delta$  może być rozpatrywany jako zmienna decyzyjna procesu optymalizacyjnego, w którym funkcja celu ma postać

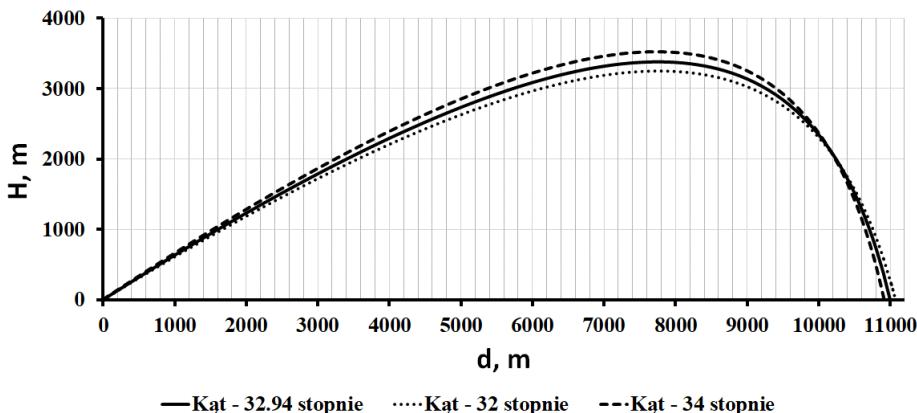
$$F_c(\delta) = |h(\delta)| \rightarrow \min \quad (3.14)$$

Mając na uwadze złożoną postać wyrażenia (3.13) oraz nieokreśloność funkcji  $f_h(\gamma)$  do znalezienia wartości kąta strzału może być wykorzystane podejście heurystyczne.

### 3.3.2 Przykład zastosowania metody AIG do wyznaczenia optymalnego kąta strzału

W celu ilustracji koncepcji algorytmu AIG, rozpatrzonego przykładowo strzelania do celu oddalonego o 11 000 m dla prędkości początkowej  $v_0 = 700$  m/s oraz współczynnika oporu powietrza  $k = 0,05$  1/s, optymalna wartość kata strzału  $\delta = 32,94^\circ$ . Skuteczność precyzyjnego trafienia do celu z tak dużej odległości jest istotnie zależna od wartości kata strzału, albowiem dla  $\delta' = 32^\circ$  pocisk przeleci 265 m nad celem, a dla  $\delta'' = 34^\circ$  upadnie ok. 100 m przed celem. Jak widać, z uwagi na opór powietrza, zwiększenie kata strzału nie powoduje w tym przypadku wydłużenia jego zasięgu (dla rzutu ukośnego bez uwzględniania oporu powietrza dla kątów od zera do  $\pi/4$  im większy kąt tym większy zasięg strzału).

Na poniższym rysunku przedstawiono krzywe balistyczne dla rozważanego przypadku, dla kąta optymalnego  $\delta_s = 32,94^\circ$  oraz kątów  $\delta' = 32^\circ$  oraz  $\delta'' = 34^\circ$ .



Rys. 3.2 Krzywe balistyczne dla trzech różnych kątów

Rozpatrując funkcję celu bez czynników zakłócających lot pocisku, należy podkreślić, że zastosowanie metod heurystycznych nie jest niezbędnie do wyznaczenia optymalnej wartości kata strzału  $\delta_s$ , a co więcej problem ten może zostać rozwiązany na pomocą innych technik obliczeniowych (np. metoda Newtona). Zastosowanie metod heurystycznych dla rozpatrywanego zagadnienia balistycznego może być traktowane jako test ich skuteczności.

Innowacyjny artylerzysta mając do dyspozycji baterię działa i możliwość oddania salwy, dobiera kat (osobno dla każdego działa) w procesie iteracyjnym, korzystając z zależności (3.9), przy czym doświadczalnie stwierdza, że najlepsze wyniki osiągnie stosując dwa mnożniki korekcyjne ( $p = 2$ ) oraz funkcje  $g_1$  i  $g_2$  o tej samej, specyficznej postaci, czyli

$$\delta^{i+1} = \delta^i \cdot g(\xi_1) \cdot g(\xi_2) \quad (3.15)$$

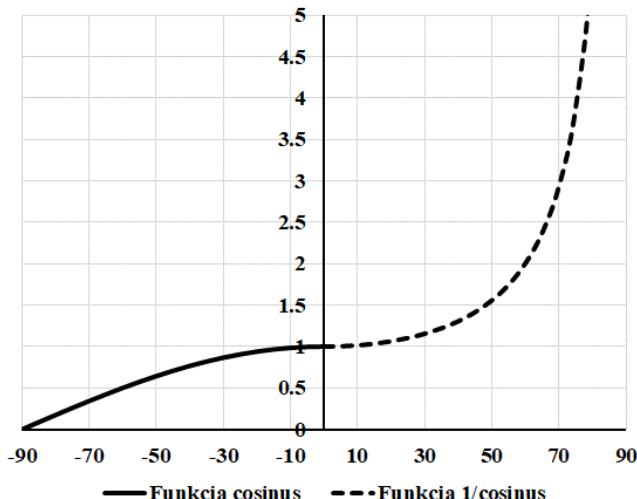
Zmienne  $\xi_1$  i  $\xi_2$  nie są związane z fizyczną teorią balistyki, aczkolwiek, ponieważ są to kąty  $\xi_1 = \alpha, \xi_2 = \beta$  wybierane losowo w procesie losowania z przedziału  $(-\alpha_{\max}, \alpha_{\max})$  oraz  $(-\beta_{\max}, \beta_{\max})$ , to nazywamy je „kątami korekcyjnymi”. Losowanie kątów  $\alpha$  oraz  $\beta$  z zaznaczonych wyżej przedziałów podlega rozkładowi normalnemu (lub równomierнемu), o wartości średniej równej zero oraz odchyleniu standardowym, spełniającym zależność  $3\sigma = \alpha_{\max} = \beta_{\max}$  (przyjmowano  $\alpha_{\max} = \beta_{\max} = \frac{\pi}{2}$ ). Kluczowe znaczenie ma wybór funkcji  $g(\alpha)$ ,  $g(\beta)$ . Chodzi bowiem o to, by dla wartości „kątów korekcyjnych”  $\alpha$  oraz  $\beta$  bliskich zeru współczynniki korygujące miały wartość bliską jeden, przy czym dla  $\alpha > 0$ ,  $g(\alpha) > 1$ , a dla  $\alpha < 0$ ,  $g(\alpha) < 1$  (analogicznie dla kąta  $\beta$ ). Dla wylosowanych z małym prawdopodobieństwem kątów bliskich  $\alpha_{\max}$  oraz  $-\alpha_{\max}$  współczynniki te powinny wyraźnie różnić się od wartości 1 ( $g(\alpha) \gg 1$ ,  $g(\alpha) \ll 1$ ), tak by zapewnić odpowiednio dużą korektę zmiennej decyzyjnej  $\delta$ . Podobnie przyjmujemy dla kąta  $\beta$ . Kierując się tymi wskazówkami, dobrano funkcję

$$\begin{aligned} g(\alpha) &= \csc(\alpha) = (\cos(\alpha))^{-1} \text{ dla } \alpha > 0 \\ g(\alpha) &= \cos(\alpha) \text{ dla } \alpha < 0 \end{aligned} \quad (3.16)$$

oraz

$$\begin{aligned} g(\beta) &= \csc(\beta) = (\cos(\beta))^{-1} \text{ dla } \beta > 0 \\ g(\beta) &= \cos(\beta) \text{ dla } \beta < 0 \end{aligned} \quad (3.17)$$

Wykres funkcji korekcyjnej pokazano na rys. 3.3.



Rys. 3.3 Przebieg funkcji korekcyjnych

Funkcje korekcyjne mogą być modyfikowane poprzez podnoszenie do odpowiedniej potęgi lub korzystanie z jednego zestawu kątów w obliczeniach. Poprawia to skuteczność algorytmu.

Tym samym ostateczna postać algorytmu AIG, dla pojedynczego działa, ma postać przedstawioną poniżej:

1. wybierz losowo kąt strzału  $\delta_0$ ,
2. wyznacz wartość funkcji celu  $F_c(\delta_0)$ , ustaw licznik iteracji (dla jednej salwy) na  $i = 1$ , podstaw  $\delta^i = \delta_0$ ,
3. wylosuj „kąt korekcyjny”  $\alpha^i$ ; dla  $\alpha^i > 0$ ,  $g(\alpha^i) = (\cos(\alpha^i))^{-1}$ ,  
dla  $\alpha^i < 0$ ,  $g(\alpha^i) = \cos(\alpha^i)$ ,
4. wylosuj „kąt korekcyjny”  $\beta^i$ ; dla  $\beta^i > 0$ ,  $g(\beta^i) = (\cos(\beta^i))^{-1}$ ,  
dla  $\beta^i < 0$ ,  $g(\beta^i) = \cos(\beta^i)$ ,
5. wyznacz skorygowaną wartość kąta strzału  $\delta^{i+1} = \delta^i \cdot g(\alpha^i) \cdot g(\beta^i)$ ,
6. wyznacz wartość funkcji celu  $F_{obj}(\delta^{i+1})$ ,
7. jeżeli  $|F_c(\delta^{i+1}) - F_c(\delta^i)| < \varepsilon$  zakończenie obliczeń – punkt 8, jeżeli nie powrót do punktu 3,
8. zakończenie obliczeń:  $\delta_s = \delta^{i+1}$  (optymalny kąt strzału).

### 3.3.3 Uogólniony algorytm AIG

Dostosowując przedstawiony wyżej algorytm dla salwy wykonanej z  $m$  działał, można przedstawić go w postaci pozwalającej na rozwiązywanie dowolnych zadań optymalizacyjnych, tak jak przedstawiono to za pomocą wzorów (3.8) i (3.9). Niezmienne jednak stosowane są dwa mnożniki korygujące ( $p = 2$ ), a ich postać nie ulega zmianie – wykorzystywana jest funkcja cosinus lub jej odwrotność (można je modyfikować poprzez podnoszenie do odpowiedniej potęgi lub korzystanie z jednego zestawu kątów w obliczeniach). Przedział wyboru losowego „kątów korygujących”  $(-\alpha_{max}, \alpha_{max})$  oraz  $(-\beta_{max}, \beta_{max})$  ulega w trakcie procesu iteracyjnego zawężeniu, zgodnie z zależnościami postępu geometrycznego

$$\begin{aligned}\alpha_{max}^i &= \alpha_{max} \cdot a^i \\ \beta_{max}^i &= \beta_{max} \cdot a^i\end{aligned}\tag{3.18}$$

przy czym liczba  $a$  należy do przedziału  $(0, 1)$ .

Ostatecznie więc uogólniona postać algorytmu AIG może być zapisana w następujący sposób:

1. losowanie wartości początkowych składowych wektora decyzyjnego

$$[\mathbf{x}_1^0] = [x_{11}^0, x_{12}^0, \dots, x_{1n}^0] \quad \text{pierwszy wektor startowy}$$

...

$$[\mathbf{x}_m^0] = [x_{m1}^0, x_{m2}^0, \dots, x_{mn}^0] \quad m - \text{ty wektor startowy}$$

2. znalezienie wartości funkcji celu dla wektora startowego zmiennych sterujących  $F_c(\mathbf{x}_1^0) \dots F_c(\mathbf{x}_m^0)$ ,  $F_{c\_best} = \min[F_c(\mathbf{x}_1^0) \dots F_c(\mathbf{x}_m^0)]$ ,

3. ustalenie licznika iteracji (dla jednej salwy) na  $i = 1$ , podstawienie  $\mathbf{x}_1^i = \mathbf{x}_1^0 \dots \mathbf{x}_m^i = \mathbf{x}_m^0$

4. ustalenie zakresu  $\alpha_{max}^i$  i  $\beta_{max}^i$ ,

5. korekta wartości składowych wektora decyzyjnego poprzez współczynniki korekcyjne  $g(\alpha), g(\beta)$ :

**rozwiązańe  $j = 1$** , losowanie „kątów korekcyjnych”  $\alpha_{11}^i \dots \alpha_{1n}^i$  i  $\beta_{11}^i \dots \beta_{1n}^i$  według zdefiniowanego rozkładu prawdopodobieństwa, korekta składowych wektora decyzyjnego

$$x_{11}^{(i+1)} = x_{11}^{(i)} \cdot g(\alpha_{11}^i) \cdot g(\beta_{11}^i) \dots x_{1n}^{(i+1)} = x_{1n}^{(i)} \cdot g(\alpha_{1n}^i) \cdot g(\beta_{1n}^i)$$

...

**rozwiązańe  $j = m$** , losowanie „kątów korekcyjnych”  $\alpha_{m1}^i \dots \alpha_{mn}^i$  oraz  $\beta_{m1}^i \dots \beta_{mn}^i$  według zdefiniowanego rozkładu prawdopodobieństwa, korekta składowych wektora decyzyjnego

$$x_{m1}^{(i+1)} = x_{m1}^{(i)} \cdot g(\alpha_{m1}^i) \cdot g(\beta_{m1}^i) \dots x_{mn}^{(i+1)} = x_{mn}^{(i)} \cdot g(\alpha_{mn}^i) \cdot g(\beta_{mn}^i),$$

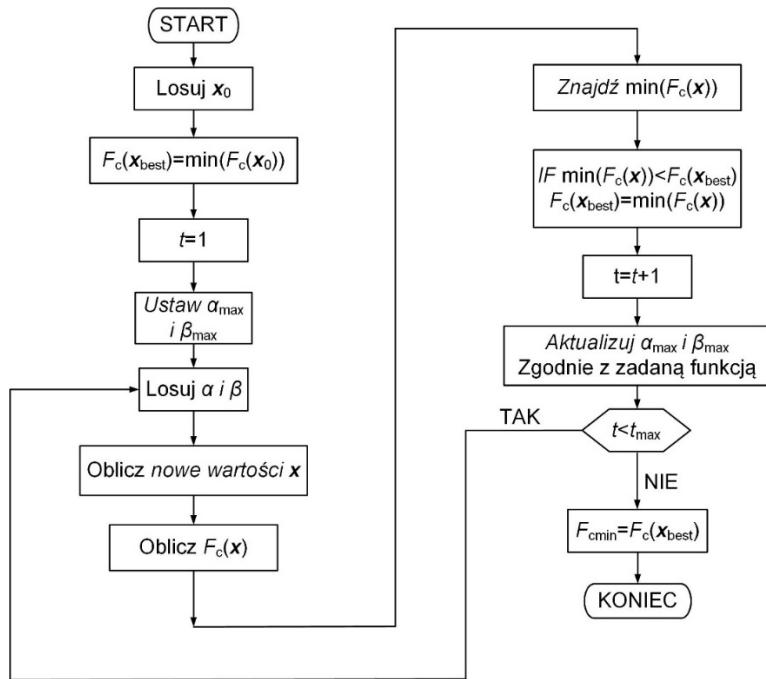
6. wyznaczenie wartości funkcji celu  $F_c(\mathbf{x}_m^{i+1})$  dla kolejnych rozwiązań  $1 \dots m$ ; sprawdzenie czy wartość funkcji celu dla kolejnego rozwiązania jest mniejsza od wartości funkcji celu z poprzedniej iteracji (jeżeli tak to należy zastąpić to rozwiązanie rozwiązaniem aktualnym – lepszym, a jeżeli nie to pozostawić), identyfikacja rozwiązania  $q$ , dla którego funkcja  $F_c(\mathbf{x}_q^{i+1}) = \min[F_c(\mathbf{x}_1^{i+1}), \dots, F_c(\mathbf{x}_m^{i+1})]$  osiąga wartość minimalną,

7. sprawdzenie czy dla zidentyfikowanego rozwiązania  $q$  zachodzi  $F_c(\mathbf{x}_q^{i+1}) < F_{c\_best}(\mathbf{x}_{best}^i)$ , jeśli tak to przyjęcie go jako kolejnego wektora decyzyjnego w procesie iteracyjnym, czyli  $\mathbf{x}_{best}^{i+1} = \mathbf{x}_q^{i+1}$ ,

8. sprawdzenie kryterium zatrzymania obliczeń ( $i+1 = i_{max}$ ), koniec (punkt 9) lub powrót do punktu 4 ( $i = i+1$ ),

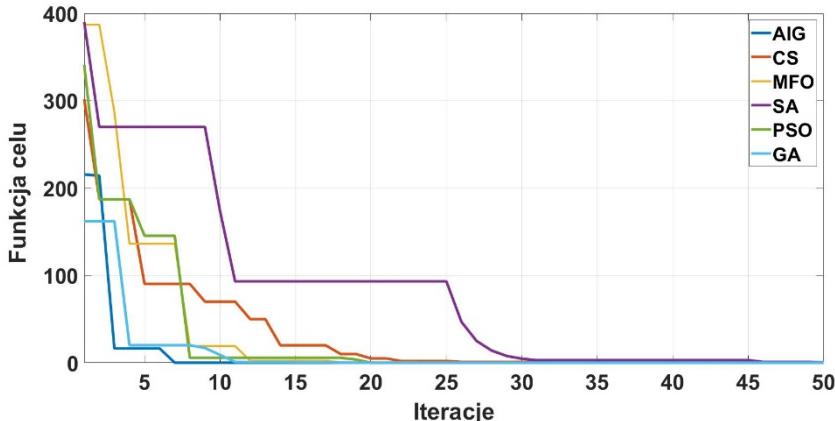
9. koniec obliczeń.

Schemat blokowy algorytmu AIG przedstawiono na rys. 3.4.



Rys. 3.4 Schemat blokowy algorytmu AIG

Na rys. 3.5 przedstawiono porównanie przebiegu najlepszych wartości funkcji celu (zależność 3.14) dla sześciu różnych metod heurystycznych (AIG, CS – przeszukiwanie kukułczane, GA – algorytmy genetyczne, MFO – algorytm optymalizacji čmy, PSO – rój częstek, SA – symulowane wyżarzanie).



Rys. 3.5 Przebieg funkcji celu dla różnych algorytmów

Dla 30 powtórzeń obliczeń uzyskano następujące wyniki (najlepszej wartości funkcji celu – BV, średniej – AVE, odchylenia standardowego – STD) w tab. 3.1:

**Tab. 3.1 Porównanie wyników obliczeń dla różnych algorytmów**

| Algorytm | Optymalna wartość zmiennej<br><i>x</i> | BV        | AVE       | STD      |
|----------|--|-----------|-----------|----------|
|          |  |           |           |          |
| AIG      | 32,9421                                | 1,819e-12 | 1,819e-12 | 0        |
| CS       | 32,9421                                | 1,819e-12 | 1,898e-11 | 8,31e-11 |
| MFO      | 32,9421                                | 1,819e-12 | 1,819e-12 | 0        |
| SA       | 32,9421                                | 6,982e-8  | 6,46      | 33,57    |
| PSO      | 32,9421                                | 1,819e-12 | 56,16     | 85,78    |
| GA       | 32,9421                                | 1,89e-4   | 74,88     | 91,7     |

Widać wyraźnie, że proponowany algorytm bardzo dobrze radzi sobie z uzyskaniem rozwiązania. Jak widać z rys. 3.5, dość szybko udaje się znaleźć poszukiwane optimum.

### 3.4 Strojenie algorytmu

#### 3.4.1 Losowanie kątów

Zmiana obydwu kątów może się odbywać według rozkładu normalnego o średniej równej 0 i odchyleniu standardowym np.  $\frac{\alpha_{\max}}{3}, \frac{\beta_{\max}}{3}$ , gdzie

$\alpha_{\max} = 90^\circ, \beta_{\max} = 90^\circ$ . Oczywiście kąty mogą być przyjmowane dowolnie. Rozkład normalny w obydwu przypadkach uzasadniony jest tym, że podczas celowania do określonego celu zakres zmian kątów nie jest stosunkowo duży. Najbardziej prawdopodobne są niewielkie wartości odchylenia, w kierunku poziomym i pionowym, w stosunku do prostej łączącej cel i lufę działa. Do zmiany obydwu kątów odpowiedni może być również rozkład równomierny, szczególnie wtedy, gdy mamy do czynienia z problemem wielowymiarowym bądź o bardzo dużej liczbie zmiennych. Podczas testowania algorytmu na różnych funkcjach celu zauważono, że dla niektórych problemów lepszym wyborem był właśnie rozkład równomierny.

#### 3.4.2 Zmiany zakresu losowania kątów

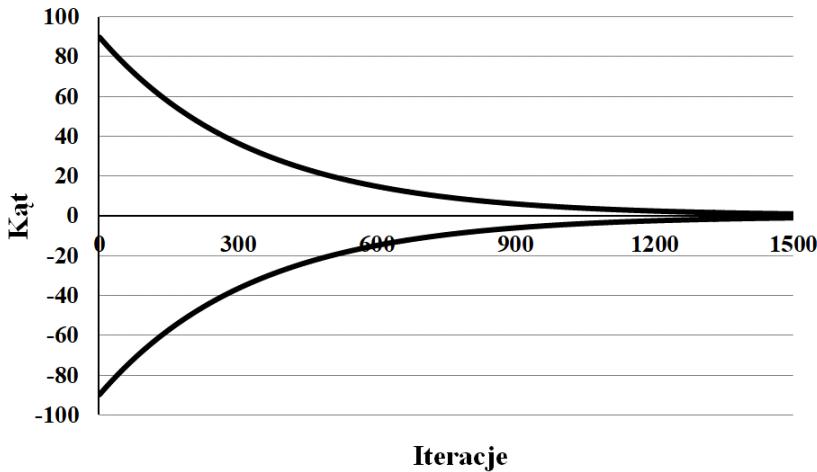
W podstawowej wersji algorytmu można założyć pełną losowość zmian kątów  $\alpha$  i  $\beta$  w dopuszczalnych granicach, od początku do końca obliczeń. Niekiedy lepsze efekty daje jednak zmniejszanie zakresu zmian kątów wraz ze wzrostem liczby iteracji ponieważ poprawia to zbieżność. Poniżej przedstawiono kilka schematów zmian:

- **geometryczny**, zgodnie ze wzorem:

$$\alpha_t = \alpha_{\max} a^t \quad (3.19)$$

gdzie

np.  $\alpha_{\max} = 90^\circ$ ,  $a$  jest liczbą z przedziału  $(0, 1)$ , a  $t$  kolejną iteracją,

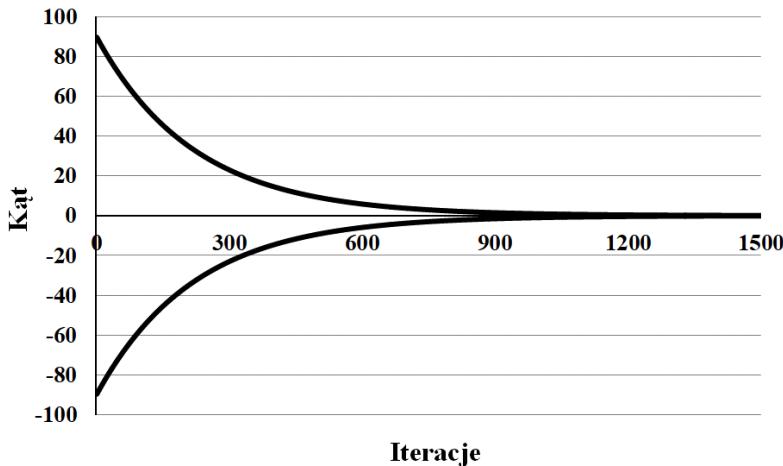


Rys. 3.6 Przykładowy, geometryczny schemat zmniejszania kątów  $\alpha$  i  $\beta$

- według funkcji eksponent, zgodnie ze wzorem:

$$\alpha_t = \alpha_{\max} \cdot e^{(-t \cdot r)} \quad (3.20)$$

gdzie np.  $\alpha_{\max} = 90^\circ$ ,  $r$  jest parametrem zmniejszającym (może być odwrotnością iloczynu liczb zmiennych i maksymalnej liczby iteracji), a  $t$  kolejną iteracją,

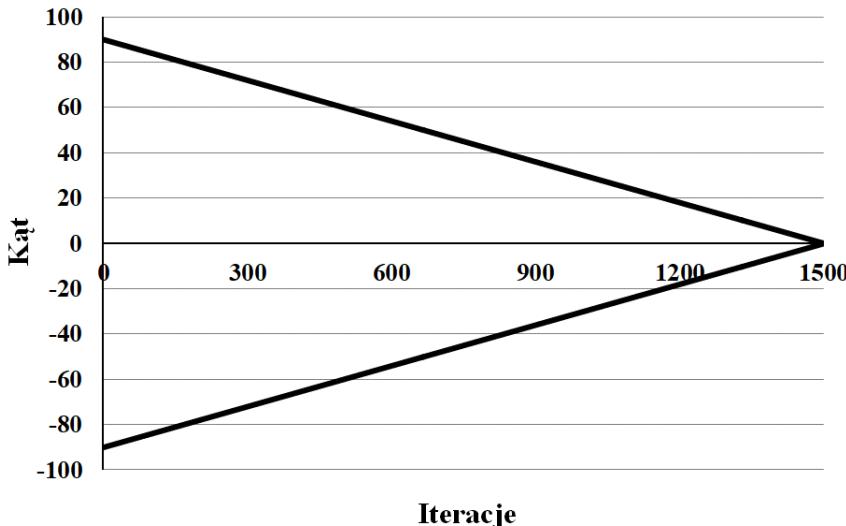


Rys. 3.7 Przykładowy schemat zmniejszania kątów  $\alpha$  i  $\beta$  według funkcji eksponent

- **liniowy**, zgodnie ze wzorem:

$$\alpha_t = \alpha_{\max} - t \cdot \frac{\alpha_{\max} - \alpha_{\min}}{N} \quad (3.21)$$

gdzie np.  $\alpha_{\max} = 90^\circ$ ,  $N$  jest liczbą iteracji,  $\alpha_{\min}$  jest wartością końcową kąta, a  $t$  kolejną iteracją,



Rys. 3.8 Przykładowy, liniowy schemat zmniejszania kątów  $\alpha$  i  $\beta$

Zmniejszanie dopuszczalnego zakresu zmian kątów  $\alpha$  i  $\beta$  uzasadnione jest zbieżnością algorytmu do punktu optymalnego. W początkowej fazie przeszukuje się całą przestrzeń, a w miarę zbliżania się do końca obliczeń, dziedzina funkcji zostaje coraz bardziej ograniczana, tak aby poprawiać wartość funkcji celu i osiągnąć optimum. Możliwe są również inne schematy zmian dopuszczalnego zakresu kątów. Wyróżnić można np. schemat logarytmiczny, według funkcji trygonometrycznych lub wykładniczych.

Skuteczność algorytmu wzrasta, jeżeli zastosuje się kilka „salw”. Sprowadza się to do tego, że po osiągnięciu maksymalnej (założonej) liczby iteracji, powtarza się cykl obliczeń, przyjmując wartości początkowe wektorów zmiennych, równe ostatnim wartościom tychże wektorów w poprzedniej „salwie”.