# Wstęp

Algorytm wyszukiwania harmonii (HS – Harmony Search) został wprowadzony w 2001 roku. Jest on inspirowany sposobem improwizacji harmonii przez muzyków jazzowych. Algorytm HS narzuca mniej wymagań matematycznych w porównaniu do innych algorytmów optymalizacji, gdyż wykorzystuje on wyszukiwanie losowe. HS generuje nowy wektor rozwiązania, po rozważeniu wszystkich istniejących wektorów, gdzie algorytm generyczny rozważa tylko dwa wektory. Algorytm ma problem z wyszukiwaniem lokalnym. W celu rozwiązania tego problemu wykorzystuje się ulepszone algorytmy wyszukiwania harmonii (IHS – Improved Harmony Search).

# Ulepszony algorytm wyszukiwania harmonii

Algorytm IHS pozwala na wyszukiwanie harmonii w o wiele wydajniejszy sposób w porównaniu ze zwykłym HS. Dzieje się to dzięki zastosowaniu parametrów algorytmu ewoluujących wraz z kolejnymi generacjami wektorów. Algorytmy HS składają się z poniższych kroków:

Krok 1. Inicjalizacja parametrów problemu i algorytmu.

Krok 2. Inicjalizacja pamięci harmonii.

Krok 3. Improwizacja nowej harmonii.

Krok 4. Aktualizacja pamięci harmonii.

Krok 5. Sprawdzenie kryterium stopu.

Kroki zostały opisane poniżej.

## Krok 1. Inicjalizacja parametrów problemu i algorytmu

W Kroku 1 należy wprowadzić następujące parametry:

Parametry problemu (zmienne *X*, *Xmin*, *Xmax* są wektorami):

* funkcja problemu – *f(X),*
* zakresy dla każdego elementu wektora wejściowego funkcji – *Xmin* i *Xmax,*

Parametry algorytmu:

* *NI* – liczba iteracji,
* *HMS* – rozmiar pamięci harmonii (Harmony Memory Size),
* *HMCRmin* i *HMCRmax* – zakres dla wskaźnika uwzględniania pamięci (Harmony Memory Consideration Rate)
* *PARmin* i *PARmax* – zakres dla wskaźnika regulacji „wysokości tonu” (Pitch Adjustment Rate)
* *BWmin* i *BWmax* – zakres dla szerokości pasma wyszukiwania harmonii (Bandwidth)

Dane te są przekazywane do programu.

## Krok 2. Inicjalizacja pamięci harmonii

Na podstawie danych wprowadzonych w Kroku 1 przez program tablica *HM* o rozmiarach   
*N* x *HMS*, gdzie *N* jest ilością elementów wektora *X* przekazanego w funkcji *f(X),* wypełniana jest losowymi zmiennymi w zakresach odpowiadających danym elementom wektora.

## Krok 3. Improwizacja nowej harmonii

W tym kroku generowany jest nowy wektor x’ = (x’1, ... x’N), bazując na trzech zasadach:

* Rozważanie pamięci,
* Dostosowywanie ”wysokości tonu”,
* Losowy wybór

Dla każdej generacji wyliczane są wskaźniki *HMCR(gn),* *PAR(gn)* i *BW(gn)* według zaproponowanych funkcji:

gdzie

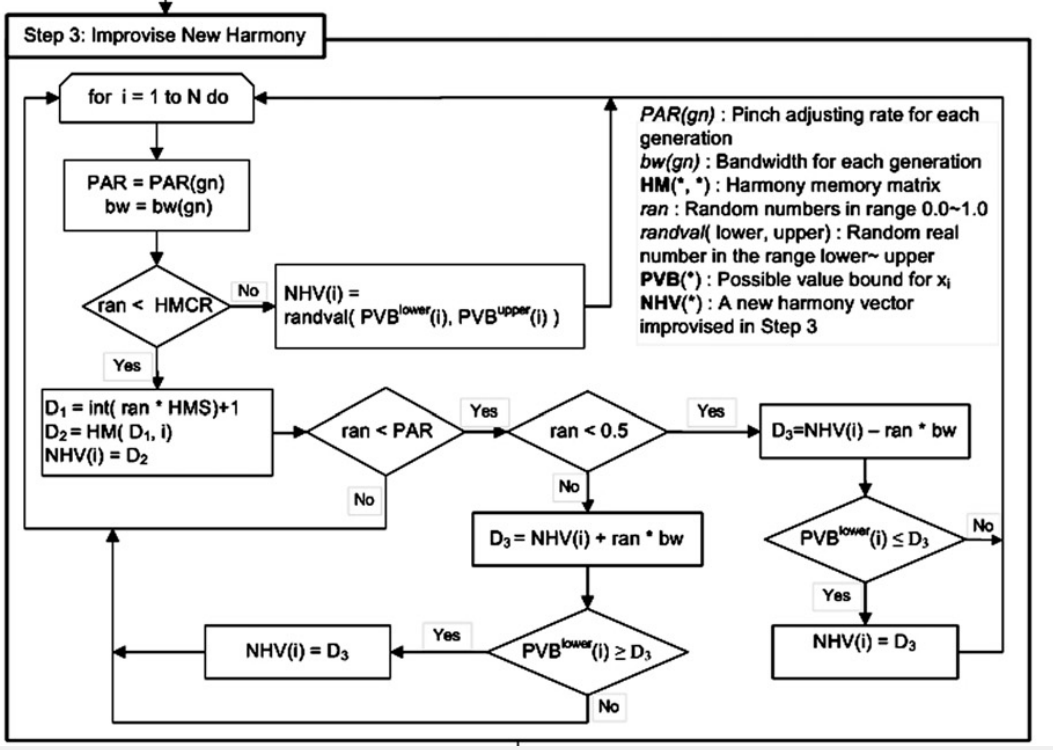
*gn­­* – generacja.

Poniżej przedstawiono wykresy zależności tych wskaźników od generacji algorytmu, dla parametrów:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | HMCR | PAR | BW | NI |
| min | 0.2 | 0.2 | 0.1 | 100 |
| max | 0.8 | 0.8 | 10 |  |

W przypadku kiedy zakres któregokolwiek parametru algorytmu jest większy od zera algorytm jest ulepszonym algorytmem wyszukiwania harmonii, a jeżeli zakresy są zerowej szerokości algorytm zachowa się jak zwykły algorytm wyszukiwania harmonii.

Nowy wektor jest generowany według algorytmu przedstawionego poniżej. Streszczając – dla każdego elementu nowego wektora rozważa się przypisanie mu losowej wartości z dziedziny z prawdopodobieństwem 1 – HMCR. W przeciwnym razie przypisywana jest mu wartość losowego elementu z pamięci. W przypadku wykorzystania elementu z pamięci rozważa się dostosowanie wartości tego elementu z prawdopodobieństwem PAR. Dostosowanie wartości polega na zmianie wartości po przepisaniu o wartość w zakresie pasma BW, pod warunkiem, że nowa wartość znajduje się w dziedzinie.



Rysunek ) To może przerysujemy

## Krok 4. Aktualizacja pamięci harmonii

W przypadku kiedy *f(X’),* gdzie *X*’ jest nowo wygenerowanym wektorem, jest rozwiązaniem lepszym od najgorszego rozwiązania w pamięci, najgorszy wektor zostaje zastąpiony nowym.

## Krok 5. Sprawdzenie kryterium stopu

Jeśli osiągnięto maksymalną liczbę iteracji algorytm jest zakańczany, a jeśli nie algorytm powraca do kroku 3 zwiększając numer generacji o 1.

# Przezentacja programu

# Porównanie wyników dla różnych parametrów.