**Algorytm „Harmony Search” - Andrzej Kowal**

# Algorytm „Harmony search”

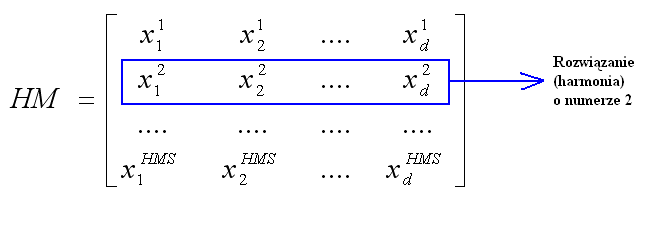
## Założenia ogólne

Algorytm o nazwie Harmony Search (HS) opisano po raz pierwszy w pracy [8]. Jego zastosowania można odnaleźć w [7], [9], [10], [11]. Został on oparty na zasadach, jakimi kierują się muzycy jazzowi podczas improwizacji. Z teoretycznego punktu widzenia, celem w tworzeniu utworu improwizowanego jest uzyskanie ciągu następujących po sobie harmonii (współdźwięków), z których każda wywiera na odbiorcy dobre wrażenie odsłuchowe. Jednocześnie harmonie te mają ściśle określoną kolejność występowania, która również podlega subiektywnej ocenie słuchacza. Muzycy pod każdym z obu powyższych względów dbają o to, aby utwór miał wysokie walory estetyczne. Proces osiągnięcia stanu, w którym utwór uznany zostaje za odpowiednio dopracowany, jest procesem iteracyjnym. Zdefiniowany w ten sposób problem stworzenia optymalnej kompozycji (pomijając treść, którą autor zamierza w utworze przekazać) może być odpowiednikiem rzeczywistego problemu optymalizacji. Muzycy dążący do uzyskania współdźwięków o wysokich walorach estetycznych mogą zostać zastąpieni meta-heurystycznym algorytmem poszukującym rozwiązania danego problemu o jak najlepszej wartości funkcji celu.

Jakkolwiek odczucia człowieka towarzyszące słuchaniu muzyki są wysoce subiektywne, już w starożytności stworzono podstawy teorii harmonii i podano matematyczne zależności pomiędzy częstotliwościami, które tworzą współbrzmienia oceniane jako przyjemne. Zwrócono na to uwagę w pracy [9]. Kolejną istotną własnością algorytmu HS jest prowadzenie poszukiwań w całym dostępnym zakresie rozwiązań, a więc rozważane jest optimum globalne problemu - analogicznie do muzyka, który ma do dyspozycji całą rozpiętość tonalną swojego instrumentu.

## Pamięć harmonii (HM) i improwizacja harmonii

Elementem kluczowym algorytmu jest pamięć harmonii (ang. *Harmony memory,* HM). Jej strukturę przedstawia Rys. 4.1. Pozostając przy analogii do zespołu muzycznego pamięć harmonii można określić jako zbiór o ustalonym rozmiarze (oznaczanym HMS, ang. *harmony memory size*), zawierający harmonie będące, zdaniem muzyków, najlepsze z dotychczas stworzonych.



Rys. .. Struktura pamięci algorytmu HS

Każda osoba w zespole zna pewien zbiór dźwięków (składowych harmonii), które jest w stanie zagrać. Jest to przeważnie cały zakres tonalny instrumentu należącego do danej osoby. Nowe współbrzmienia są generowane poprzez zagranie przez każdego z muzyków jednego z dźwięków dostępnych w ich pamięci indywidualnej. Bardzo ważny dla całego procesu improwizacji jest nie tyle sposób oceny nowo powstałej harmonii, lecz dobór dźwięków w trakcie jej tworzenia. Muzyk wybierając dźwięk do nowo tworzonego współbrzmienia (czyli w trakcie improwizacji nowej harmonii) przeważnie korzysta z jednej z trzech możliwości :

1. Ton wybierany jest z HM. Muzyk wie, który dźwięk w każdej z harmonii był zagrany przez niego. W ten sposób przeglądając wszystkie dostępne współdźwięki, jest on w stanie stwierdzić, który z dźwięków zagranych przez niego pojawiał się najczęściej, lub też, który dawał najlepsze rezultaty. Ostateczny wybór jest więc subiektywny.
2. Ton wybierany jest z HM zgodnie z regułą 1, po czym jest nieznacznie modyfikowany. Modyfikacja polega przeważnie na zagrania dźwięku o częstotliwości zbliżonej do dźwięku pierwotnego. Możliwy jest również wybór harmonicznej (np. kwarty lub kwinty) dźwięku pierwotnego.
3. Ton wybierany jest losowo z całej pamięci indywidualnej (czyli z zakresu tonalnego instrumentu danego muzyka).

Takie postępowanie przeprowadza każda osoba w zespole, co daje w rezultacie nową harmonię. Wygenerowany współdźwięk jest oceniany przez zespół i, o ile jest lepszy niż najgorszy z już zawartych w pamięci, wprowadzany jest do HM zamiast harmonii o najgorszej ocenie. W ten sposób rozmiar HM pozostaje niezmienny.

## Algorytm HS w zastosowaniu do problemów optymalizacji

Mechanizm opisany w punkcie 4.2 można, poprzez proste analogie, przebudować tak, aby umożliwiał on rozwiązanie problemu optymalizacji. Mając dany problem oraz funkcję celu, która jest miarą jakości rozwiązania, postępuje się następująco :

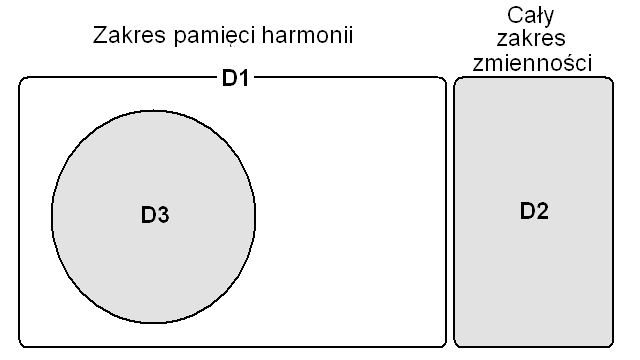
* Każdemu członkowi zespołu odpowiada jedna zmienna decyzyjna. Oznacza to, że zakres tonalny instrumentu tego muzyka odwzorowany jest w zbiór wartości zmiennej decyzyjnej mu przyporządkowanej.
* Jednej harmonii (współdźwiękowi stworzonemu przez cały zespół) odpowiada jedno rozwiązanie problemu.
* Każda harmonia ma przyporządkowaną subiektywną ocenę zespołu. Ocenie tej odpowiada wartość funkcji celu dla danego rozwiązania.
* Pamięć zawiera zbiór rozwiązań, dla których wartości funkcji celu są najlepsze spośród wszystkich wygenerowanych do tej pory rozwiązań.
* Generacja nowego rozwiązania prowadzona jest na bazie rozwiązań zawartych w pamięci i jest analogiczna do improwizacji nowej harmonii.

Oczywiste jest, że inicjalizacja algorytmu będzie polegać na zapełnieniu pamięci losowymi rozwiązaniami początkowymi. Należy jeszcze zdefiniować zasady procesu generacji kolejnego rozwiązania na bazie HM, czyli odpowiednika improwizacji nowej harmonii. Zasady te podobne są do reguł 1,2,3 zawartych w punkcie 4.2. W przypadku algorytmu HS wprowadza się dwa parametry, które decydują o sposobie wyboru zmiennej decyzyjnej wchodzącej do nowego rozwiązania. Parametry te oznacza się symbolami HMCR (ang. *harmony memory consideration ratio -* współczynnik wyboru tonu z pamięci) oraz PAR (ang. *pitch adjustment ratio -* współczynnik dostosowania tonu). Oba parametry są prawdopodobieństwami, w związku z czym ich wartości leżą w przedziale [0, 1]. Definicje reguł wyboru zmiennych decyzyjnych są następujące :

* 1. Wartość zmiennej decyzyjnej jest wybierana ze zbioru wartości, które przyjmuje ta zmienna w rozwiązaniach utrzymywanych w HM – ta operacja wykonywana jest z prawdopodobieństwem równym HMCR.
  2. Wartość zmiennej decyzyjnej jest losowana z całego dopuszczalnego zbioru wartości tej zmiennej z prawdopodobieństwem (1-HMCR).
  3. Jeśli zmienna została ustalona na podstawie reguły 1, jest ona modyfikowana (dostosowywana) z prawdopodobieństwem równym PAR, co oznacza, że reguła ta stosowana jest z prawdopodobieństwem równym HMCR\*PAR.

Jak podaje [9], modyfikacja w regule 3 polega na dodaniu do wartości ustalonej zgodnie z regułą 1 składnika przy czym =U[-b;b] gdzie b jest ustalonym parametrem algorytmu, natomiast U[-b;b] jest liczbą wylosowaną za pomocą generatora licz pseudolosowych o rozkładzie jednostajnym na przedziale [-b;b]. Takie postępowanie jest słuszne dla problemów określonych na iloczynie kartezjańskim *n* zbiorów liczb rzeczywistych czyli na przestrzeni Rn. W zadaniach dyskretnych modyfikacja polega na wyborze wartości leżącej w sąsiedztwie wartości pierwotnej. Sąsiedztwo dwóch wartości będzie zależeć od sposobu zdefiniowania odległości w rozważanej przestrzeni.

Graficzna interpretacja powyższych reguł zilustrowana jest na Rys. 4.2.



Rys. .. Obszary podjęcia decyzji o zastosowaniu reguł wyboru wartości wchodzącej do nowo tworzonego rozwiązania. Obszar Dk odpowiada regule k-tej

Prawdopodobieństwa wyboru reguł można zdefiniować następująco :

* Reguła 1 : P(D1) = HMCR
* Reguła 2 : P(D2) = 1 – HMCR
* Reguła 3 : P(D3) = HMCR \* PAR

Kryterium stopu algorytmu jest przeważnie dopuszczalna ilość iteracji oznaczana zwykle przez NI (ang. *number of improvisations*). W tej chwili możliwe jest podanie kolejnych kroków całej procedury HS :

Krok 1: Inicjalizacja parametrów algorytmu (HMS, HMCR, PAR, b, NI).

Krok 2: Inicjalizacja pamięci harmonii – generacja rozwiązań początkowych.

Krok 3: Improwizacja nowego rozwiązania na podstawie reguł 1..3 i parametrów algorytmu.

Krok 4: Obliczenie wartości funkcji celu nowego rozwiązania i aktualizacja HM w przypadku, gdy nowa wartość funkcji jest lepsza niż najgorsza z zawartych w pamięci. W takim wypadku najgorsze rozwiązanie z HM jest zastępowane ostatnio wygenerowanym. Dla zadań minimalizacji pamięć jest sortowana według rosnących wartości funkcji celu, natomiast dla maksymalizacji według wartości malejących.

Krok 5: Powtórzenie kroków 3 i 4 w przypadku gdy nie jest spełnione kryterium stopu.

Diagram ilustrujący powyższą procedurę przedstawiono na Rys. 4.3.

**Krok 1:**

- inicjalizacja parametrów

(HMS, HMCR, PAR, b, NI)

**Krok 2 :**

- inicjalizacja HM

- sortowanie HM według

rosnących wartości ocen

rozwiązań

**Krok 3 :**

- improwizacja nowego rozwiązania na podstawie HM, parametrów algorytmu oraz reguł wyboru

Nowe rozwiązanie jest lepsze od najgorszego z zawartych w pamięci ?

**Krok 5 :**

Spełniony jest warunek zatrzymania algorytmu ?

STOP

**Krok 4 :**

- modyfikacja zawartości pamięci

Tak h

Nie

Tak h

Nie

Rys. .. Diagram decyzyjny dla procedury optymalizacyjnej opartej na algorytmie HS

# Przykład – optymalizacja funkcji dwóch zmiennych za pomocą algorytmu HS

W celu lepszego zobrazowania działania omawianego algorytmu, poniżej opisano przebieg procedury rozwiązywania problemu minimalizacji funkcji testowej określonej wzorem (8).

 (8)

Jest to funkcja o 6 minimach lokalnych, z których 2 są jednocześnie globalnymi. Dzięki tej własności często wykorzystuje się ją w testach algorytmów optymalizacji, ze względu na to, że rezultaty procedur opierających się na informacji gradientowej zależą od wyboru punktu startowego. Minima globalne zlokalizowane są w punktach x’ = (-0,08984; 0,71266) oraz x” = (0,08984; -0,71266), przy czym f(x’) = f(x”) = -1,0316285.

Poniższy przykład zaczerpnięto z pracy [10]. Przyjęto następujące parametry algorytmu:

HMS = 10

HMCR = 0.85

PAR = 0.45

Zakres zmienności x1 oraz x2 wynosił [-10; 10]. Harmonią w tym zadaniu jest jedno rozwiązanie problemu czyli para liczb (x1, x2). Zawartość pamięci zainicjowanej losowymi rozwiązaniami przedstawia Tabela 1.

Tabela . Zawartość pamięci harmonii algorytmu HS dla zadania minimalizacji funkcji 2 zmiennych po inicjalizacji (na podstawie [10])

|  |  |  |  |
| --- | --- | --- | --- |
| Harmonia | x1 | x2 | f(x) |
| 1 | 3,18 | -0,40 | 169,95 |
| 2 | -6,60 | 5,08 | 26274,83 |
| 3 | 6,66 | 7,43 | 37334,24 |
| 4 | 6,76 | 8,32 | 46694,70 |
| 5 | -7,58 | 5,56 | 60352,77 |
| 6 | 7,76 | 4,70 | 67662,40 |
| 7 | -8,25 | 2,75 | 95865,20 |
| 8 | -8,30 | 8,53 | 120137,09 |
| 9 | -9,01 | -8,05 | 182180,00 |
| 10 | -9,50 | 3,33 | 228704,72 |

Rozwiązania uporządkowane są według rosnących wartości funkcji celu. Najlepsze rozwiązanie znajduje się w wierszu o numerze 1. W każdej iteracji tworzone jest nowe rozwiązanie. Wartość pierwszej zmiennej, czyli x1, jest wybierana zgodnie z 3 regułami podanymi w podrozdziale 4.3. Reguła 1 bez modyfikacji podanej w regule 2 wybierana jest z prawdopodobieństwem równym HMCR\*(1-PAR) = 0,85\*0,55=0,4675. W takim wypadku x1 przyjmie wartość wylosowaną spośród liczb zawartych w kolumnie zmiennej x1 (Tabela 1). Prawdopodobieństwo zastosowania połączonych reguł 1 i 2 wynosi HMCR\*PAR = 0,85\*0,45 = 0,3825. Wówczas liczba wybrana z kolumny x1 będzie zmodyfikowana o wartość U[-b; b]. Ostatecznie możliwy jest całkowicie losowy wybór x1 z przedziału [-10; 10]. Ma to miejsce z prawdopodobieństwem równym 1-HMCR = 0,15. Identycznie postępuje się dla zmiennej x2. Po ustaleniu nowego rozwiązania x\* wyznaczana jest wartość f(x\*). Przykładowo w pierwszej iteracji dla wygenerowanego rozwiązania x\*=(3,183; 8,666) wartość funkcji celu f(x\*)=22454,67. Z danych znajdujących się w Tabela 1, wynika, że x\* umieszczone będzie w drugim wierszu pamięci ponieważ wartość f(x\*) leży pomiędzy wartościami dla 1 i 2 rozwiązania. Jednocześnie usunięte z HM zostanie rozwiązanie najgorsze x=(-9,500; 3,333). Zawartość pamięci po 4870 iteracjach przedstawia Tabela 2.

Tabela . Zawartość HM po 4870 iteracjach (na podstawie [10])

|  |  |  |  |
| --- | --- | --- | --- |
| Harmonia | x1 | x2 | f(x) |
| 1 | 0,08984 | -0,71269 | -1,0316285 |
| 2 | 0,09000 | -0,71269 | -1,0316284 |
| 3 | 0,09000 | -0,71277 | -1,0316283 |
| 4 | 0,09013 | -0,71269 | -1,0316281 |
| 5 | 0,08951 | -0,71269 | -1,0316280 |
| 6 | 0,08951 | -0,71277 | -1,0316279 |
| 7 | 0,08951 | -0,71279 | -1,0316278 |
| 8 | 0,09628 | -0,71269 | -1,0316277 |
| 9 | 0,08980 | -0,71300 | -1,0316275 |
| 10 | 0,09000 | -0,71300 | -1,0316274 |

Znaleziona wartość jest równa wartości optymalnej z dokładnością do 7 miejsca dziesiętnego. Należy zaznaczyć tu, że rozwiązania początkowe zostały wybrane całkowicie losowo, a mechanizmy zastosowane w algorytmie HS wymagają w tym konkretnym przypadku bardzo niewielkich nakładów obliczeniowych.