Metaheuristics
- Harmony Search -

Hyunsoo Lee

References

- X.S. Yang, “Harmony Search as a Metaheuristic Algorithm”, Music-Inspired Harmony Search Algorithm: Theory and Applications (Editor Z.W. Geem), Studies in Computational Intelligence, Springer, Vol. 191, 1-14, 2009

Background

- J.S. Yoon or S. Yoon

Origin

- 2001, ZW. Kim

- Zongwoo Kim
  - Prof. at Gacheon U.
  - Pos. Doc, J. H. U.
  - Civil Engineering, Korea U
  - Civil Engineering, CAU

- Episode

Aesthetic Quality of Music (1)

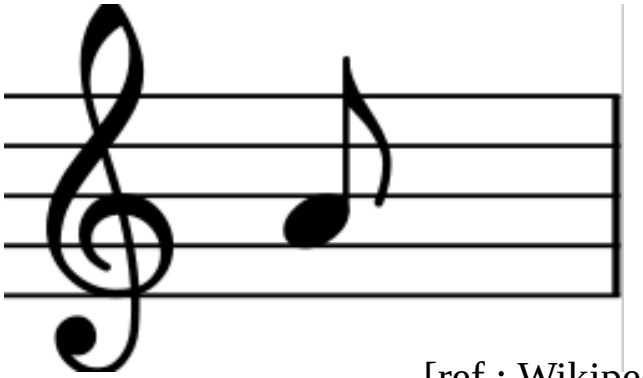
- Components
  - Pitch / Frequency

$$p_n = 69 + 12 \log_2 \left(\frac{f}{440 Hz} \right)$$

  - Timbre / Sound Quality
  - Amplitude / Loudness

Aesthetic Quality of Music (2)

- Note



[ref : Wikipedia]

$$p_n \propto Octave$$

HS (1/3)

- Improvising Scenario
  - Copy masterpiece's pitches

    → Usage of harmony memory
  - Play Similar pieces

    → Pitch adjusting
  - Compose new or Random

    → Randomization

HS (2/3)

- 1) Usages of harmony memory

$$r_{accept} \in [0,1]$$

HS (3/3)

- 2) Pitch Adjusting

$$r_{pa} \in [0.1, 0.5]$$

  - Nonlinear adjusting
  - Linear adjusting

$$x_{new} = x_{old} + b_{range} \cdot \varepsilon$$

$$\varepsilon \in [-1, 1]$$

Pseudo code (1/2)

```
Begin
        Objective function f(x)
        Generate initial harmonics (real number arrays)
        Define pitch adjusting rate(), pitch limits and bandwidth
        Define harmony Memory accepting rate
        while (t<Max number of Iterations)
                Generate new harmonics by accepting best harmonies
                if (rand>r_accept) choose an existing harmonic randomly
                else if (rand>r_pa) adjust the pitch randomly within limits
                else generate new harmonics via randomization
                end if
                Accept the new harmonics (solutions) if better
        end while
        Find the current best solutions
End
```

Pseudo code (2/2)

- Analysis

$$P_{random} = 1 - r_{accept} - r_{pa}$$

$$P_{pitch} = r_{accept} - r_{pa}$$

Limitations

- Pros
- Cons