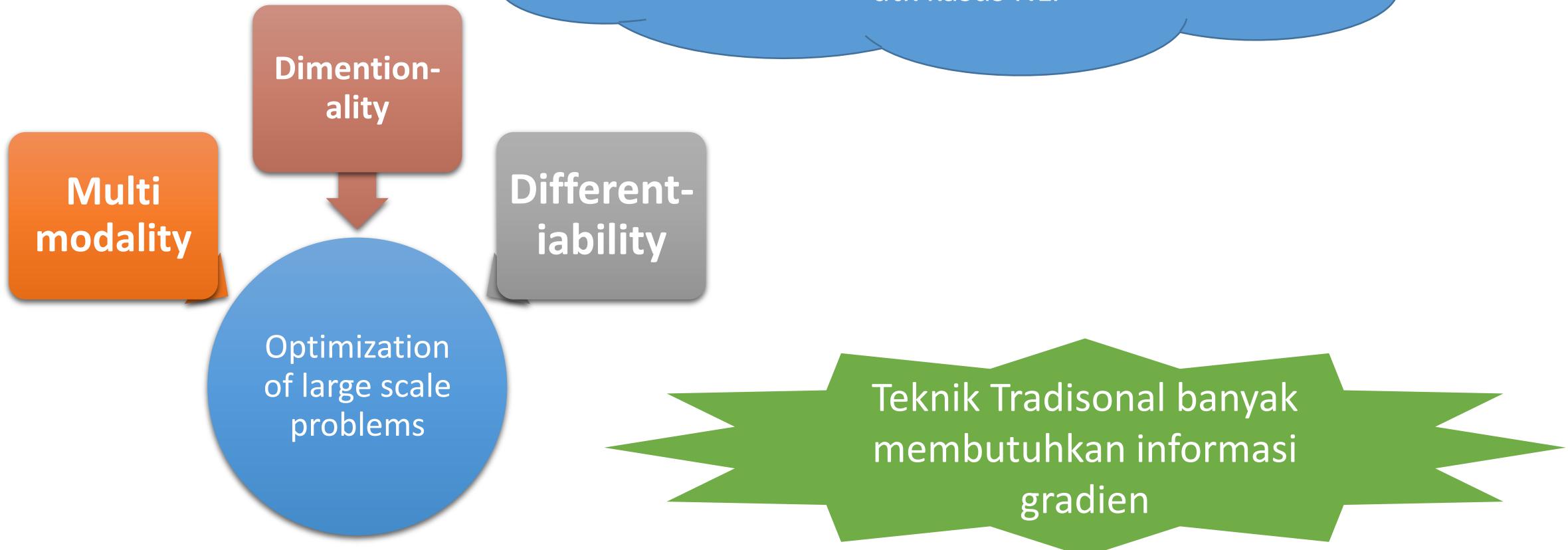




Advanced  
Optimization  
techniques

# Pengantar



# Advanced Optimization Techniques

Genetic  
Algorithm

Artificial  
Immune  
Algorithm

Differential  
Evolution

Biogeography-  
based  
Optimization

Particle  
Swarm  
Optimization

Artificial Bee  
colony  
algorithm

Harmony  
Element  
Algorithm

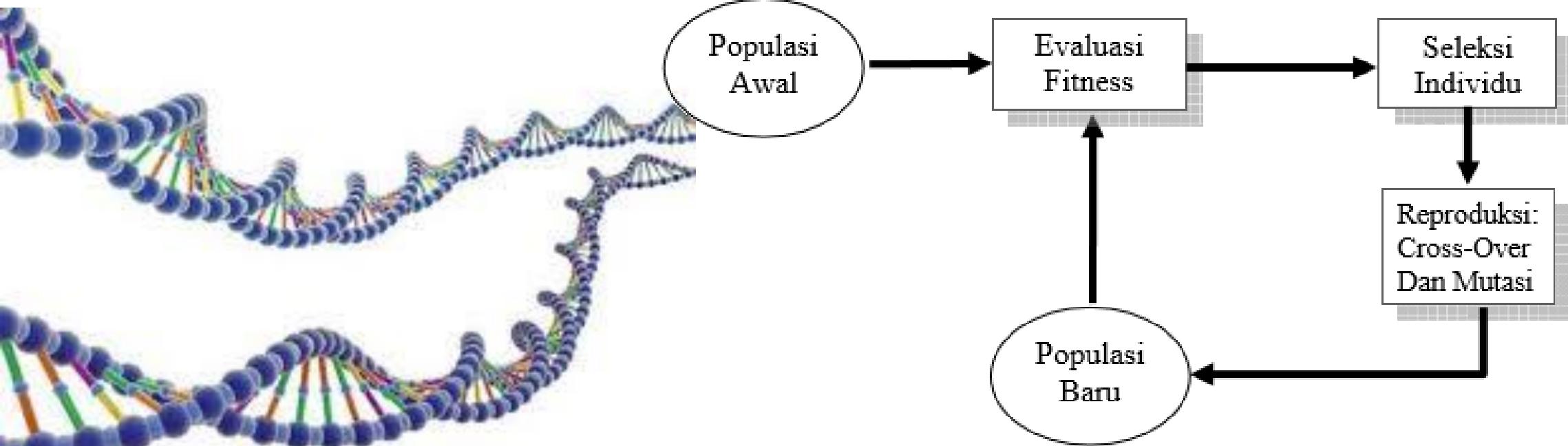
Hybird  
Algorithm

Shuffled frog  
leaping  
algorithm

Grenade  
explosion  
algorithm

# Genetic Algorithm

- Genetic algorithm didasarkan pada prinsip-prinsip genetika dan seleksi alam
- Algoritma ini banyak dipakai dalam penyelesaian masalah kombinatorial seperti TSP, VRP hingga permasalahan kontrol.

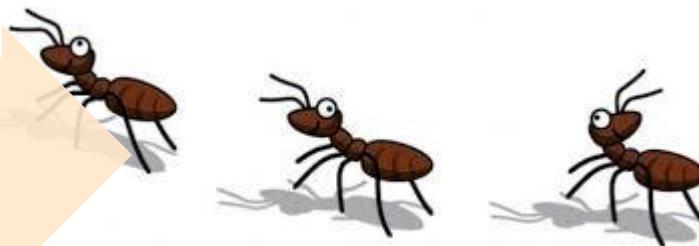


# Swarm Intelligence

- swarm intelligence (SI) dapat diartikan sebagai paradigma penyebaran kecerdasan yang inovatif untuk menyelesaikan permasalahan optimasi dengan **mengambil inspirasi dari contoh biologis**, seperti **fenomena kelompok atau kawanan (swarm) pada hewan**.

Ant Colony  
Optimization

- Berdasarkan perilaku semut mencari makanan



Particle Swarm  
Optimization

- Berdasarkan Perilaku kawanan burung



# Simulated annealing

- Simulated Annealing termasuk algoritma yang meniru perilaku fisik berupa pendinginan baja.
- Teknik ini meniru perilaku baja yang mengalami pemanasan sampai suhu tertentu kemudian didinginkan secara perlahan.

Mulai dengan vektor  $x_0$ ,  
Tentukan temperatur awal dan  
parameter lain :  $T_0, c, n$

Hitung nilai  $f_0 = f(x_0)$   
Tetapkan iterasi  $i = 1$   
Jumlah siklus  $p = 1$

Bangkitkan vektor baru  $x_{i+1}$  di  
sekitar  $x_i$ , dan hitung  $f(x_{i+1})$ .  
Hitung  $\nabla f = f_{i+1} - f_i$ .

Terima atau tolak vektor baru  
 $x_{i+1}$  sebagai solusi dengan  
kriteria Metropolis.  
Update iterasi  $i = i + 1$

Apakah  $i \geq n$

tidak

ya

Update iterasi  $p = p + 1$   
Update iterasi  $i = 1$

Kurangi temperatur

tidak

Stopping criteria  
tercapai?

ya

stop

# Cross Entropy

- Metoda CE termasuk dalam keluarga teknik Monte Carlo yang bisa digunakan untuk menyelesaikan kasus estimasi maupun optimasi.

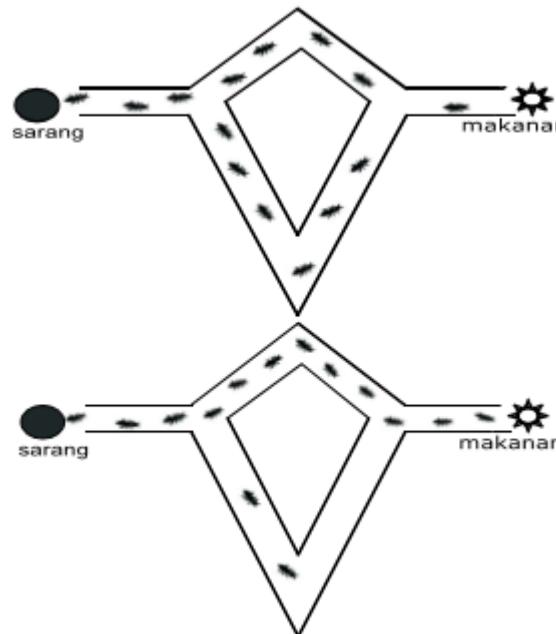
## Algoritma

1. Tentukan parameter awal  $\hat{v} = u$ ,  $\alpha$ , dan  $\rho$ . Tetapkan iterasi  $it = 1$ .
2. Bangkitkan sampel random  $X_1, , X_N$  dari fungsi probabilitas distribusi tertentu  $f(—; u)$  dan pilih sampel  $(1 - \rho)$  quantile dari performansi setelah diurutkan.
3. Gunakan sampel yang sama untuk memperbarui nilai parameter
4. Aplikasikan persamaan 7.2 untuk memuluskan vektor  $\hat{v} = u$ . Kembali ke langkah 2 dengan nilai parameter yang baru, tetapkan  $it = it + 1$ .  
$$\tilde{v}_t = \alpha \tilde{w}_t + (1 - \alpha) \hat{v}_{t-1}$$

$\tilde{w}_t$  adalah parameter vektor solusi,  
 $\alpha$  sebagai parameter smoothing,  
dengan  $0.7 < \alpha < 1$ .
5. Jika stopping criteria sudah dipenuhi, berhenti.

# Ant Colony Optimization

- ACO biasanya digunakan untuk menyelesaikan ***discrete optimization problems*** dan biasanya digunakan untuk menyelesaikan persoalan yang kompleks dimana terdapat banyak variabel.

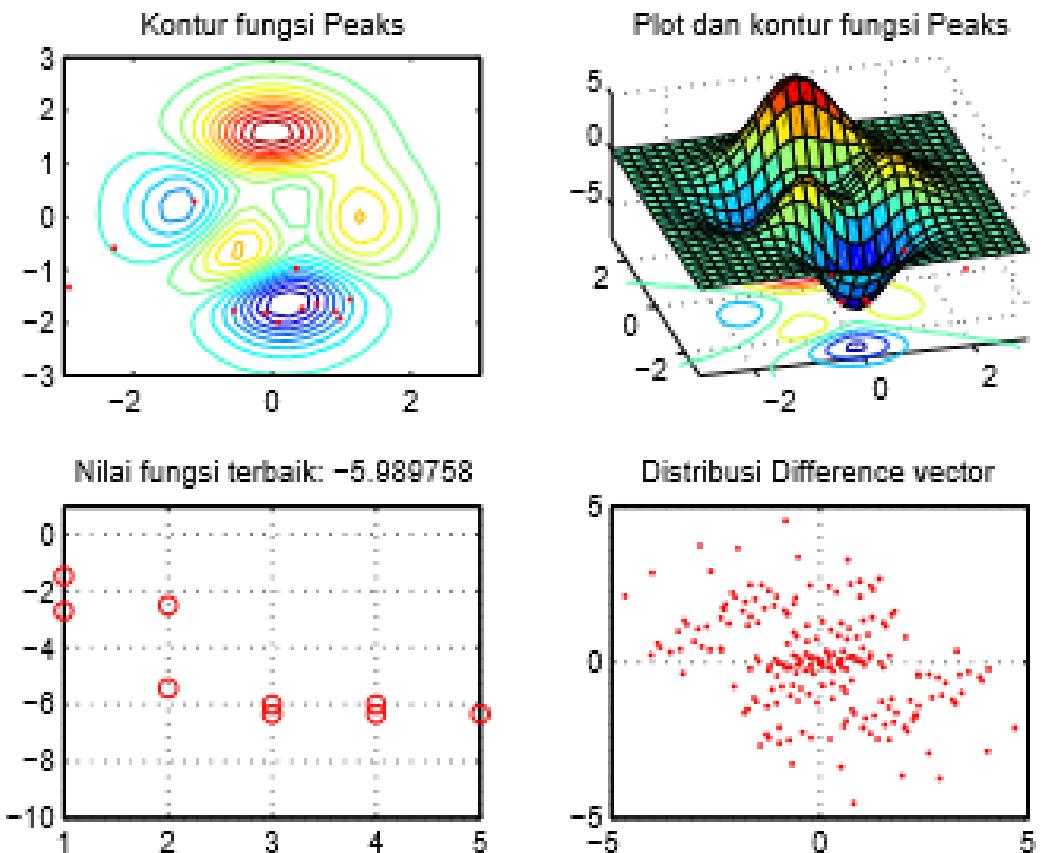


```
Input: jumlah semut,matrik jarak,itmax  
Output: tour terbaik,total jarak  
%Tahap Inisialisasi  
foreach i=1:jumlah semut do  
     $\tau(i,j) = \tau_0$   
    Hitung nilai visibility  $h(i,j) = 1/jarak(i,j)$   
end  
while it < itmax do  
    Tempatkan semua semut pada kota 1  
    foreach i=1:jumlah semut do  
        Mulailah dari kota r  
        Hitung tingkat pheromone untuk setiap ruas yang terhubung  
        dari r  
        Hitung probabilitas transisi dari kota r ke kota lain  
        Bangkitkan bilangan random  
        Gunakan roulette wheel selection untuk memilih kota  
        berikutnya  
    end  
    foreach i=1:jumlah semut do  
        Hitung jarak tour untuk setiap semut  
        Update nilai  $\tau$  menggunakan persamaan (9.10)  
        % Untuk  $\Delta\tau$  bisa digunakan  $1/f$  dari setiap semut  
        % dimana f adalah panjang tour dikurangi min panjang tour  
    end  
end
```

Algorithm 3: Algoritma ACO untuk TSP

# Differential Evolution

- Ide awal DE berasal dari Genetic Annealing algorithm
- Genetic Annealing adalah algoritma yang berdasarkan populasi (population-based) suatu algoritma untuk combinatorial optimization yang mengimplementasikan kriteria annealing melalui penerapan thresholds.



# Harmony Search

- Harmony search (HS) termasuk pendekatan metaheuristik yang mendasarkan algoritmanya pada musik.
- Setiap alat musik berkaitan dengan variabel keputusan; nada musik berkaitan dengan nilai variabel; harmoni berhubungan dengan vektor solusi.
- Beberapa contoh penerapannya adalah pada optimasi fungsi, jaringan distribusi air, pemodelan air tanah energy-saving dispatch, truss design, vehicle routing

Input: fungsi tujuan,  $itmax$

Output:  $x$  terbaik, nilai fungsi tujuan

% Tahap Inisialisasi

Tentukan HMCR ( $0.7 - 0.95$ )

Tentukan ukuran HMS %Ukuran sampel mis. 20

Tentukan pitch adjusting rate ( $PAR = 0.3$ )

Tentukan pitch bandwidth  $b, x_L$  dan  $x_U$

Bangkitkan Harmony Memory secara random sebanyak HMS  
while  $it < itmax$  do

foreach  $i=1:jumlah\ variabel$  do

Bangkitkan bilangan random  $r \approx U(0, 1)$

if  $r \leq HMCR$  then

Pilih satu nilai,  $x_{old_i}$  dari HM untuk variabel i

Bangkitkan bilangan random  $r \approx U(0, 1)$

if  $r \leq PAR$  then

Lakukan pengaturan nilai  $x_{new_i} = x_{old_i} \pm b_i$

Jika  $x_{new_i} \leq x_L$  maka  $x_{new_i} = x_L$

Jika  $x_{new_i} \geq x_U$  maka  $x_{new_i} = x_U$

end

else

| Bangkitkan vektor solusi baru  $x_{new_i}$

end

end

Bandingkan  $f(x_{new})$  dengan  $f(x_{worst})$  di dalam HM

Jika lebih baik, terima  $x_{new}$  dan simpan dalam HM

end

Temukan solusi terbaik dan fungsi obyektifnya dari HM

**Algorithm 5:** Algoritma Harmony Search



Thank You  
Thank You  
Thank You!!!!