Teknik Lokalisasi pada Jaringan Sensor Nirkabel

Bagian III (Proses Refinement Posisi)

Prima Kristalina

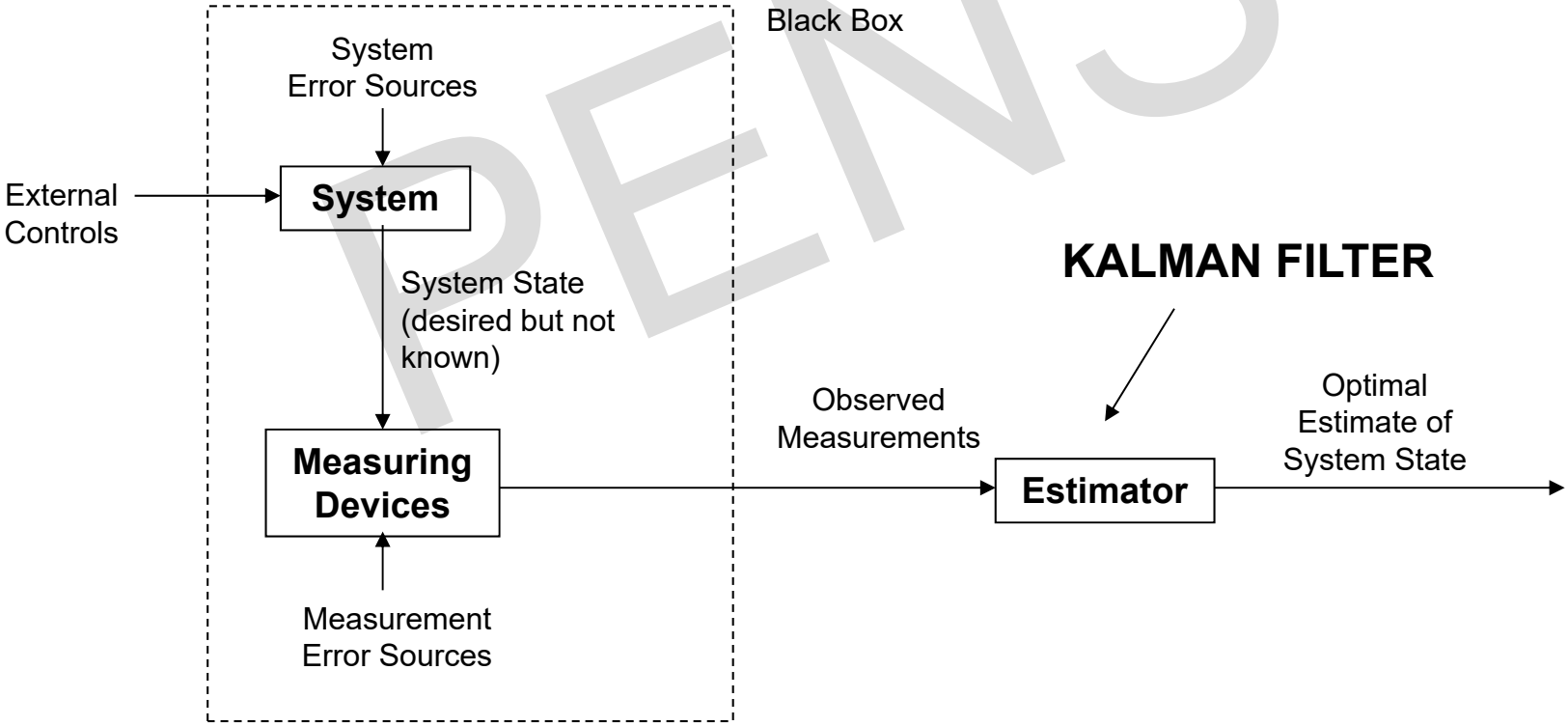
2017

Overview

- Untuk memperbaiki hasil estimasi posisi yang telah didapatkan pada part 2
- Beberapa metode/algoritma yang bisa digunakan pada bagian ini:
  - Kalman Filter
  - Particle Filter
  - Extende Kalman Filter
  - Geometric Dilution of Precision
  - Algoritma-algoritma Heuristic (Simulated Annealing, Genetic Algorithm)

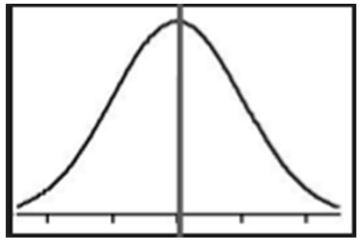
Refinement dengan Kalman Filter (1)

- Biasanya dipakai untuk perbaikan hasil estimasi jarak antar node dari pengukuran sebelumnya.
- Penyelesaian untuk sistim linier



Refinement dengan Kalman Filter (2)

- Kalman Filter: sekeleompok persamaan matematikan
- Iterative, proses recursive
- Algoritma pemrosesan data optimal dalam kriteria tertentu
- Mengestimasi kondisi sebelumnya, saat ini dan yang akan datang
- Dengan asumsi-asumsi tertentu, KF akan optimal dalam melakukan estimasi atau prediksi sekelompok data:
  - Data linier
  - Model distribusi Gaussian



Mean=median=maks

Refinement dengan Kalman Filter (3)

- Langkah-langkah di dalam Kalman Filter:

1. Initial conditions ($\hat{y}_{k-1}$ and σ_{k-1})
2. Prediksi ($\hat{y}^{-}_{k}$, σ^{-}_{k})
   - Gunakan initial conditions dan pemodelan (mis: konstanta kerapatan) untuk membuat prediksi
3. Pengukuran (z_k)
   - Lakukan pengukuran
4. Koreksi ($\hat{y}_k$, σ_k)
   - Gunakan pengukuran untuk mengkoreksi prediksi dengan "memcampur" prediksi dan residu – menggabungkan 2 distribusi Gaussian
   - Estimasi optimal jika memiliki variance lebih kecil

Refinement dengan Kalman Filter (4)

- Process to be estimated:

$y_k = Ay_{k-1} + Bu_k + w_{k-1}$ Process Noise (w) with covariance Q

$z_k = Hy_k + v_k$ Measurement Noise (v) with covariance R

- Kalman Filter

Predicted: $\hat{y}^-_k$ is estimate based on measurements at previous time-steps

$$\hat{y}^-_k = Ay_{k-1} + Bu_k$$

$$P^-_k = AP_{k-1}A^T + Q$$

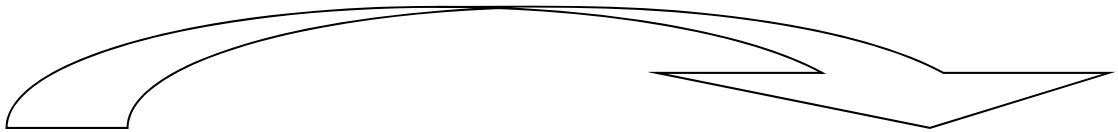
Corrected: $\hat{y}_k$ has additional information – the measurement at time k

$$\hat{y}_k = \hat{y}^-_k + K(z_k - H\,\hat{y}^-_k)$$

$$K = P^-_k H^T (HP^-_k H^T + R)^{-1}$$

$$P_k = (I - KH)P^-_k$$

Refinement dengan Kalman Filter (5)



Prediction (Time Update)

(1) Project the state ahead

$$\hat{y}^-_k = Ay_{k-1} + Bu_k$$

(2) Project the error covariance ahead

$$P^-_k = AP_{k-1}A^T + Q$$

Correction (Measurement Update)

(1) Compute the Kalman Gain

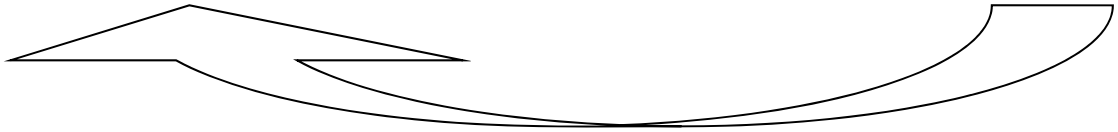
$$K = P^-_k H^T (HP^-_k H^T + R)^{-1}$$

(2) Update estimate with measurement z_k

$$\hat{y}_k = \hat{y}^-_k + K(z_k - H\,\hat{y}^-_k)$$

(3) Update Error Covariance

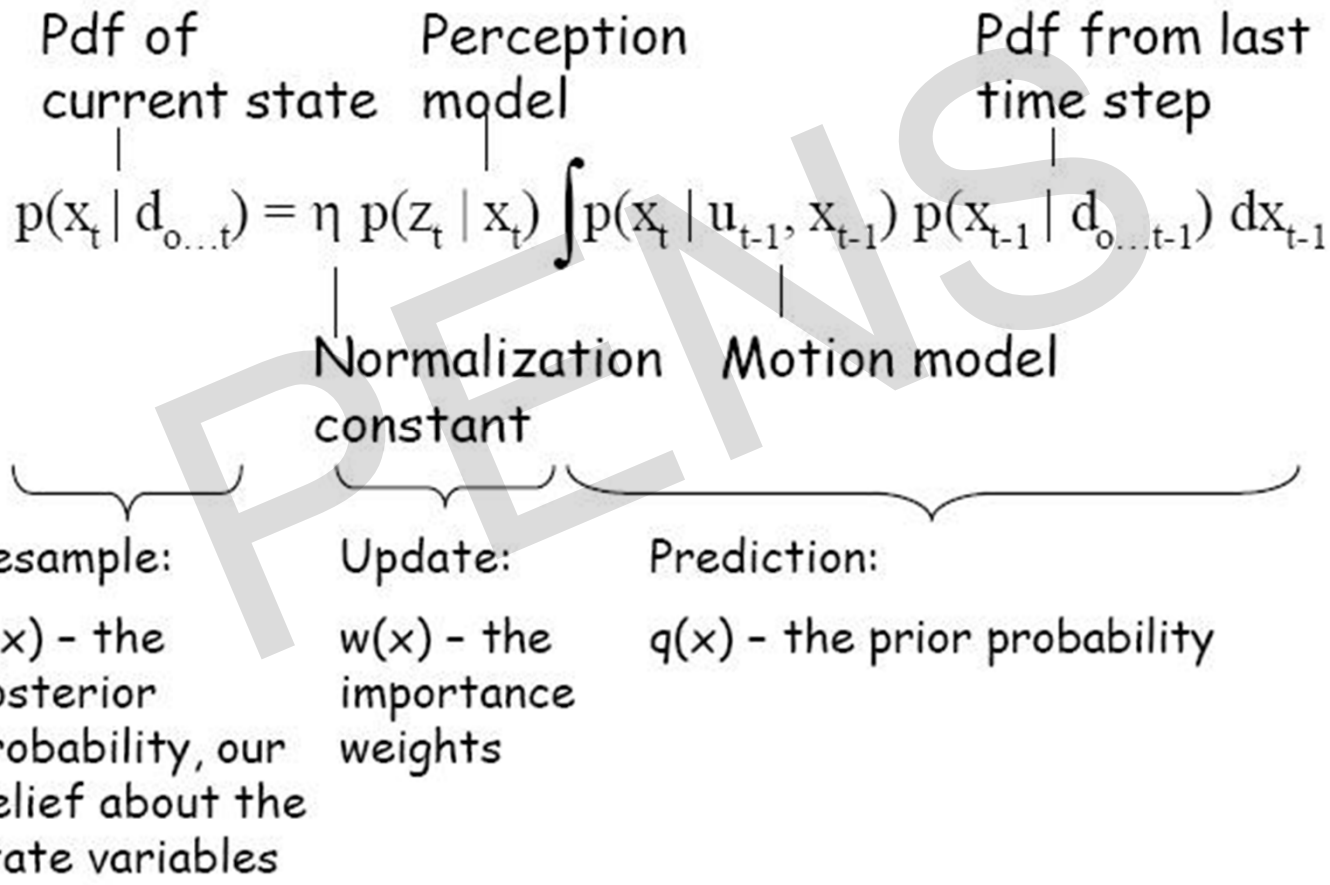
$$P_k = (I - KH)P^-_k$$



Refinement dengan Particle Filter (1)

- Untuk men-track kondisi dari sistim dinamis
- Menggunakan model Bayessian
- Men-track "belief state"
- Particle filter dipilih untuk solusi pemecahan masalah dimensi besar, yang tidak bisa diselesaikan dengan KF
- Tiga fase yang dilakukan pada PF: resample, update, prediksi

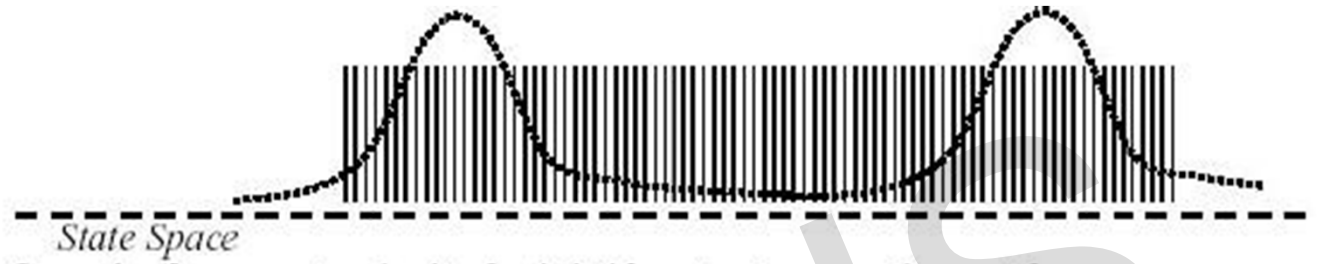
Refinement dengan Particle Filter (2)



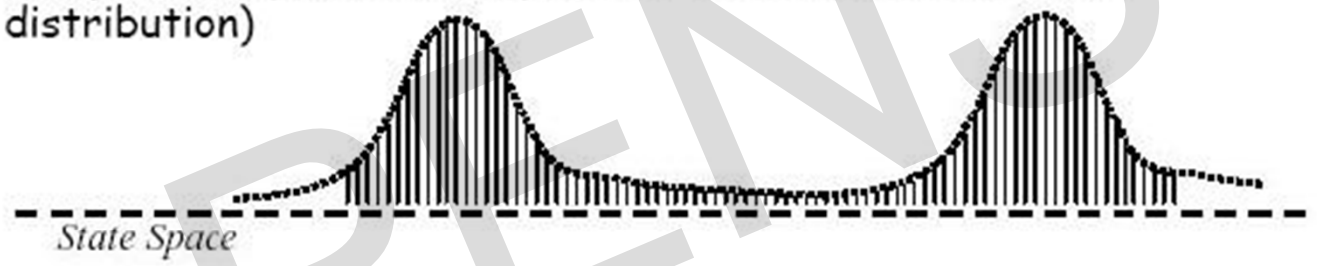
Refinement dengan Particle Filter (3)

1. Pada fase prediksi, ambil setiap partikel, tambahkan sebuah sampel random dari model gerak (motion model)
2. Pada fase update, berikan pembobotan (weighted) yang sama dengan probabilitas hasil observasi sensor dari state partikel tersebut
3. Pada fase resample, sebuah himpunan partikel baru telah terpilih, sehingga setiap partikel berusaha bergabung sesuai dengan bobot yang dimilikinya

Refinement dengan Particle Filter (4)



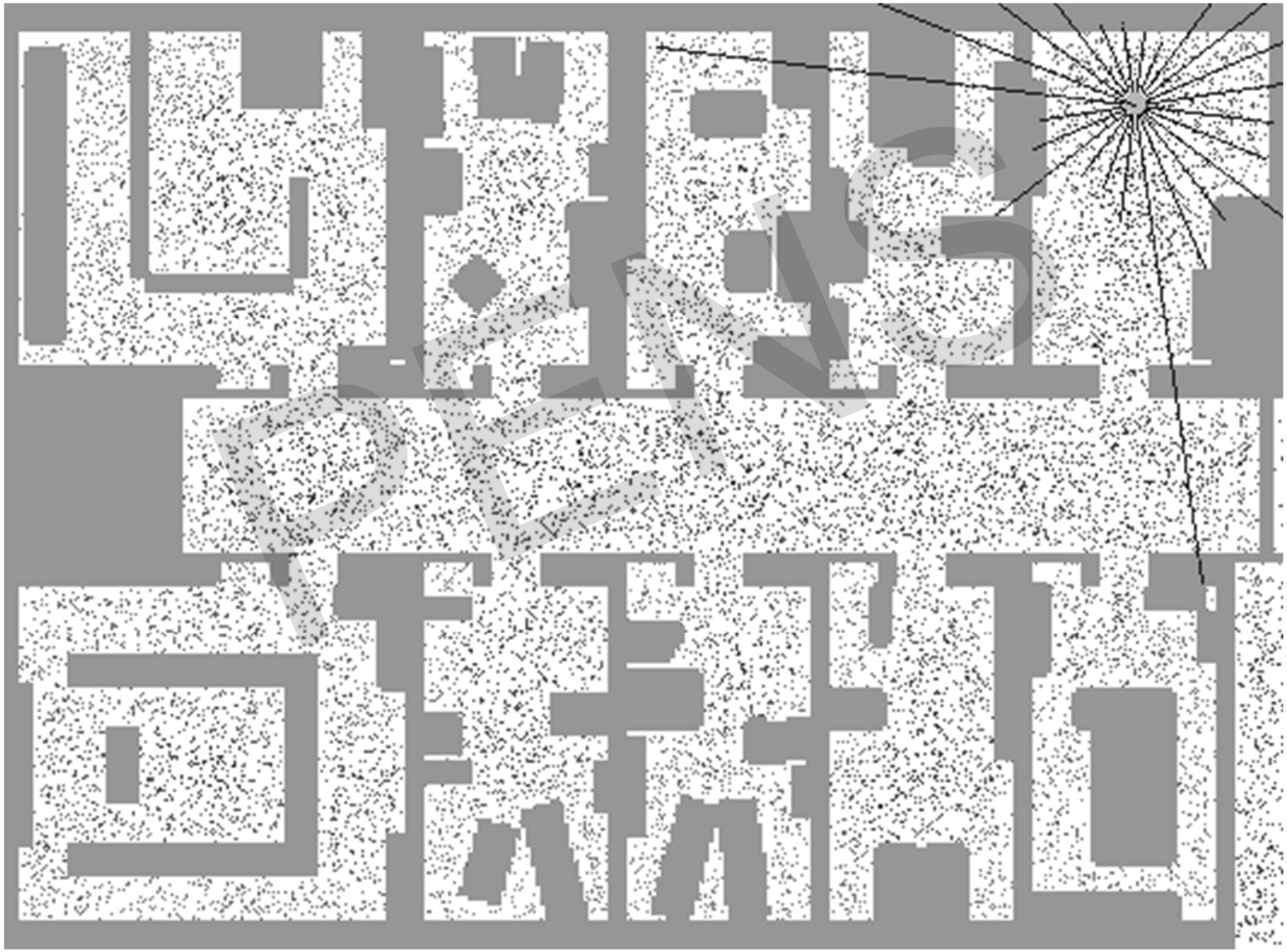
Sample from prior belief q(x) (for instance, the uniform distribution)



Compute importance weights, w(x) = p(x) /q(x)

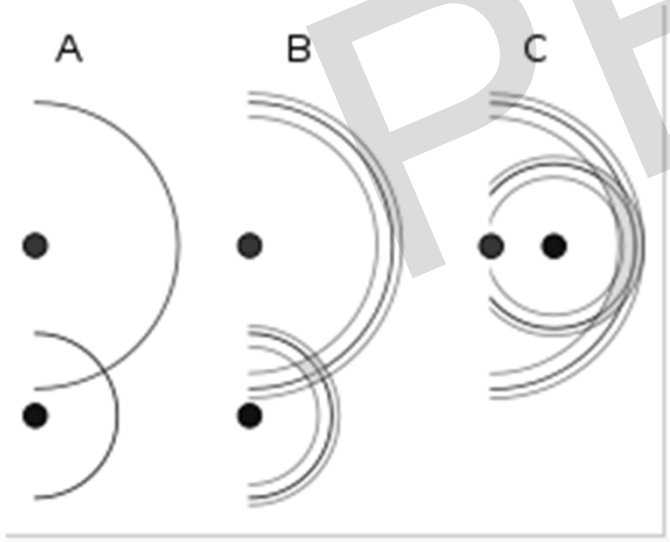
Resample particles according to importance weights to get p(x)
Samples with high weights chosen many times; density reflects pdf

Refinement dengan Particle Filter (5)



Refinement dengan GDOP (1)

DOP (Dilution of Precision) atau GDOP adalah istilah yang digunakan dalam navigasi satelit dan Teknik geomatika untuk menyatakan efek multiplikasi dari navigasi geometri satelit terhadap kepresisian pengukuran posisi obyek



Pada gambar A, sesorang mengukur jarak dirinya terhadap dua landmark (dlm hal ini 2 lingkaran) dan memposisikan dirinya berada pada interseksi/perpotongan dari 2 lingkaran tsb)
Pada gambar B, pengkuruan jarak memiliki beberapa error bound, sehingga lokai seseorang berada di sebuah tempat di sekitar daerah hijau
Pada gambar C, masih dgn error bound, namun error disebabkan karena peletakan landmark, bukan karena pengukuran

Refinement dengan GDOP (2)

Pada penerimaan sinyal satelit, dikatakan baik apabila posisi satelit-satelit saling berjauhan, sehingga nilai DOP kecil (kanan), dan apabila satelit saling berdekatan akan menghasilkan nilai DOP besar, sehingga memperburuk kalkulasi posisi estimasi

Navigation satellites with poor geometry for Geometric Dilution of Precision (GDOP).

Navigation satellites with good geometry for Geometric Dilution of Precision (GDOP).

Refinement dengan GDOP (3)

Jarak satelit-pusat bumi, s dan jarak user –pusat bumi, u, sehingga jarak satelit-user, r:

$$\mathbf{r = s - u}$$

Kesalahan posisi sebuah satelit j terhadap user dinyatakan sebagai:

$$r_j = \sqrt{\left(x_j - x_u\right)^2 + \left(y_j - y_u\right)^2 + \left(z_j - z_u\right)^2}$$

Matriks A untuk kesalahan pengukuran 4 Satelit terhadap sebuah user:

$$A = \begin{bmatrix} \frac{(x_1 - x_u)}{R_1} & \frac{(y_1 - y_u)}{R_1} & \frac{(z_1 - z_u)}{R_1} & -1 \\ \frac{(x_2 - x_u)}{R_2} & \frac{(y_2 - y_u)}{R_2} & \frac{(z_2 - z_u)}{R_2} & -1 \\ \frac{(x_3 - x_u)}{R_3} & \frac{(y_3 - y_u)}{R_3} & \frac{(z_3 - z_u)}{R_3} & -1 \\ \frac{(x_4 - x_u)}{R_4} & \frac{(y_4 - y_u)}{R_4} & \frac{(z_4 - z_u)}{R_4} & -1 \end{bmatrix}$$

Refinement dengan GDOP (4)

Matriks Q:

$$Q = \left(A^T A \right)^{-1} \text{ dan}$$

$$Q = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xz} & \sigma_{xt} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{yz} & \sigma_{yt} \\ \sigma_{xz} & \sigma_{yz} & \sigma_z^2 & \sigma_{zt} \\ \sigma_{xt} & \sigma_{yt} & \sigma_{zt} & \sigma_t^2 \end{bmatrix}$$

Nilai GDOP:

$$PDOP = \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2}$$

$$TDOP = \sqrt{\sigma_t^2}$$

$$GDOP = \sqrt{PDOP^2 + TDOP^2}$$

PDOP = Position (3D) DOP
TDOP = Time DOP

Refinement dengan Genetic Algorithm (1)

Sebuah GENETIC ALGORITHM dibuat untuk menyelesaikan populasi kandidat dari sebuah masalah, dan diselesaikan secara iterative menggunakan sekumpulan operator stokastik

Refinement dengan Genetic Algorithm (2)

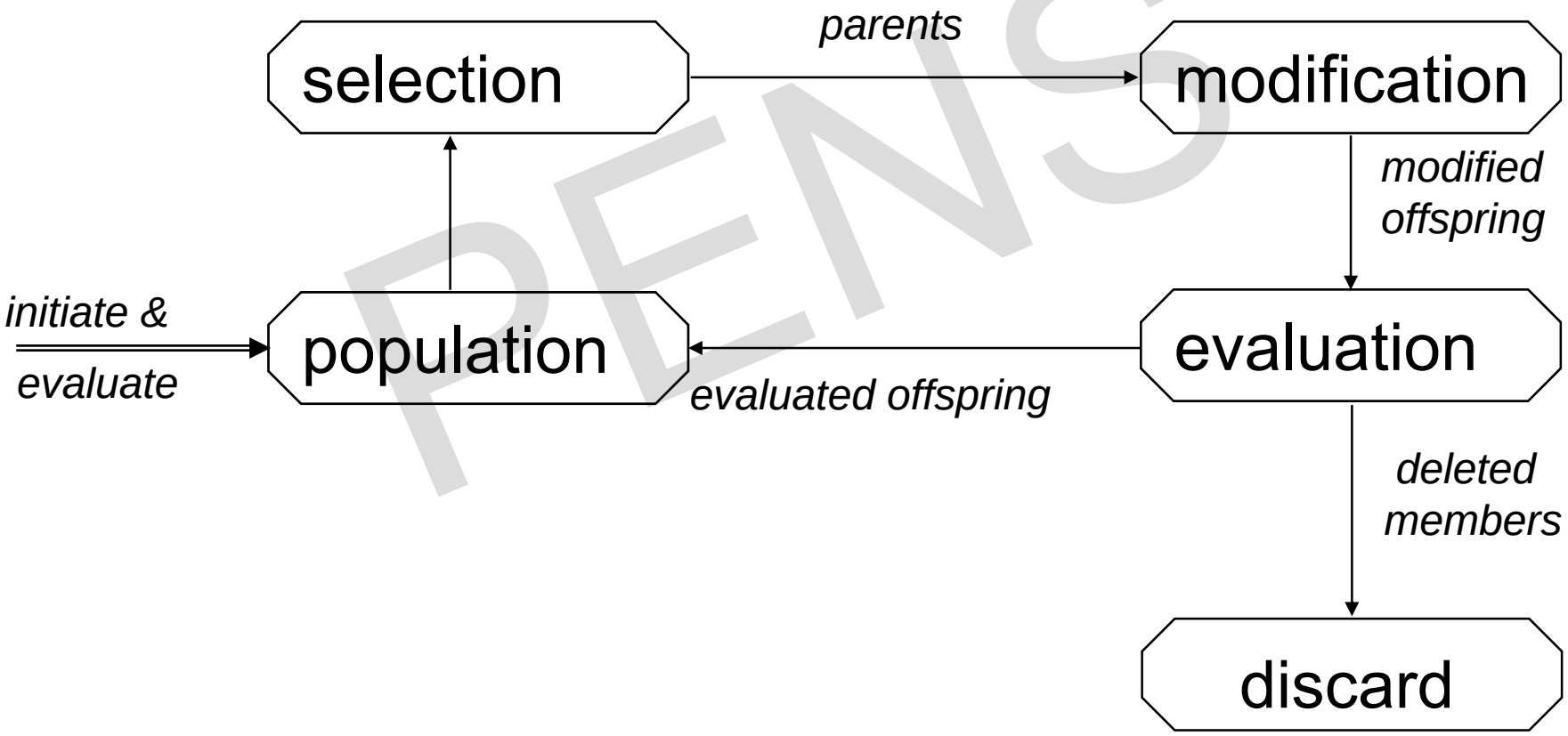
Genetic Algorithm	**Nature**
Kumpulan solusi	Populasi organisme (species)
Stochastic operators	Selection, recombination and mutation dalam proses evolusi
Secara iterative menjalankan sekelompok operator stokastik untuk menyelesaikan masalah	Evolusi dari populasi untuk mempertahankan diri pada lingkungannya

Refinement dengan Genetic Algorithm (3)

- ***OPERATOR STOKASTIK:***
- ***Selection*** mereplika solusi-solusi yang berhasil ditemukan pada populasi dengan kecepatan yang sebanding dengan kualitas relative nya.
- ***Recombination*** men-dekomposisi dua solusi berdekatan dan secara random mencampurnya untuk membentuk solusi yang baru
- ***Mutation*** secara random mengganggu solusi candidate

Refinement dengan Genetic Algorithm (4)

Siklus Evolusi:



Refinement dengan Genetic Algorithm (5)

Table 1: Pseudocode of genetic algorithm

Step 1: Initialization
The number of individuals, NIND;
The maximum number of generation, MAXGEN;
The precision of variables, PRECI;
The generation gap, GGAP;
Initialized the times of generation gen= 0;

Step 2: Coding
Initialize the group by Gray, P(t);

Step 3: Evaluate
Evaluate the fitness of each individual in the group P(t);

Step 4: Genetic Iteration
While t < MAXGEN
  Selecting the individuals for Crossover;
  Crossing the selected individuals by certain probability;
  Mutating the individuals of group by certain probability;
  Evaluating the fitness of the new group;
  Producing a new group after the evolution, P(t);
  Partial Best = min(Evaluate P(t));
  Update the times of generation, t = t + 1;
End

Step 5: Obtain the result
Global Best = Partial Best;

Refinement dengan Simulated Annealing(1)

SIMULATED ANEALING merupakan sebuah proses pembekuan dari besi yang dipanaskan

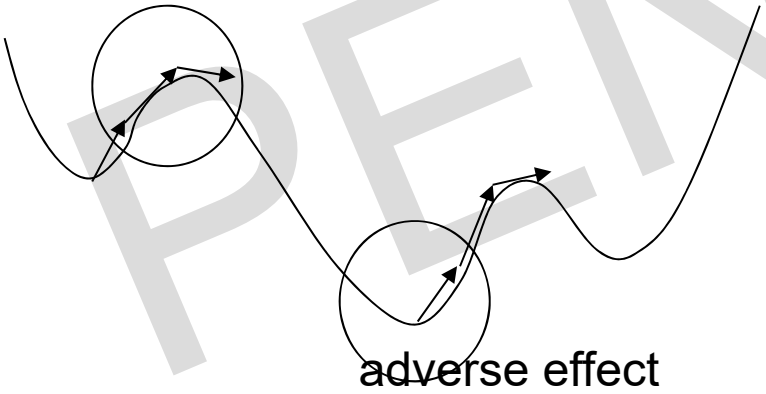
<u>IDE:</u>

Dengan membiarkan pendakian sesekali dalam proses pembekuan, ada kemungkinan mendapatkan kesempatan untuk menghindari kondisi local minimal

Refinement dengan Simulated Annealing(2)

desired effect

Help escaping the local optima.



adverse effect

Might pass global optima after reaching it

Refinement dengan Simulated Annealing(4)

Pengontrolan Proses Annealing

Penerimaan step pencarian (Metropolis Criterion):

- Assume the performance change in the search direction is Δ
- Always accept a descending step, i.e. $\Delta \leq 0$
- Accept a ascending step only if it pass a random test, $\exp(-\Delta/T) > random[0,1)$

Refinement dengan Simulated Annealing(3)

Pengontrolan Proses Annealing

Jadwal Pendinginan:

- T, the *annealing temperature*, is the parameter that control the frequency of acceptance of ascending steps.
- We gradually reduce temperature $T(k)$.
- At each temperature, search is allowed to proceed for a certain number of steps, $L(k)$.
- The choice of parameters $\{T(k), L(k)\}$ is called the *cooling schedule*.

Refinement dengan Simulated Annealing(4)

Algoritma SA

0) k = 0;

1) Search (i $\rightarrow$ j), performance difference Δ;

2) If $\Delta \leq 0$ then accept, else
if exp(-Δ/T(k)) > random[0,1) then accept;

3) Repeat 1) and 2) for L(k) steps;

4) k = k+1;

5) Repeat 1) – 4) until stopping criterion is met.

Refinement dengan Simulated Annealing(5)

Pseudocode Algoritma SA

```
1. Create random initial solution γ
2. E_old=cost(γ);
3. for(temp=temp_max; temp>=temp_min;temp=next_temp(temp) ) {
4.         for(i=0;i<i_max; i++ ) {
5.                 succesor_func(γ); //this is a randomized function
6.                 E_new=cost(γ);
7.                 delta=E_new-E_old;
8.                 if(delta>0)
9.                         if(random() >= exp(-delta/K*temp);
10.                                undo_func(γ); //rejected bad move
11.                        else
12.                                E_old=E_new //accepted bad move
13.                else
14.                        E_old=E_new; //always accept good moves
           }
      }
```