

Local Search & Optimization Problem

Chastine Fatichah
Departemen Teknik Informatika
Maret 2023



IF

Capaian Pembelajaran Matakuliah

Mahasiswa mampu menjelaskan, mengidentifikasi, merancang, dan menerapkan *intelligent agent* untuk *problem* yang sesuai dengan memanfaatkan algoritma pencarian yang meliputi *uninformed search*, *informed search*, *heuristic search*, *adversarial search*, serta algoritma *search* untuk *Constraint Satisfaction Problem*

Pokok Bahasan

- Local search & Optimization problem
- Hill climbing
- Simulated Annealing
- Genetic Algorithm



Teknik Pencarian (Search)

1. Pencarian tanpa informasi (*uninformed search*)
 2. Pencarian dengan informasi (*informed search*)
 3. Pencarian dengan optimasi (*local search & optimization*)
 4. Pencarian dengan informasi status lawan (*adversarial search*)
 5. Pencarian dengan batasan kondisi (*constraint satisfaction problems*)
-
- Teknik 1 dan Teknik 2 mencari jalur (*path*) status solusi dari *initial state* sampai *goal state*
 - Teknik 3 hanya membutuhkan *state* yang memenuhi kondisi final



IF

Local Search & Optimization Problem

- Hanya butuh *state* yang memenuhi kondisi final
 - Contoh: solusi problem 8-queens → posisi 8 bidak dengan jumlah bidak saling menyerang minimal
 - Solusi adalah konfigurasi akhir 8 bidak
 - Tidak perlu tahu urutan bidak yang diletakkan di papan
 - Contoh: solusi problem pencarian jarak terpendek yang membutuhkan urutan jalur dari kota awal ke kota akhir
 - Solusi adalah konfigurasi urutan kota yang dikunjungi
 - Tidak perlu melakukan eksplorasi setiap node atau kota



IF

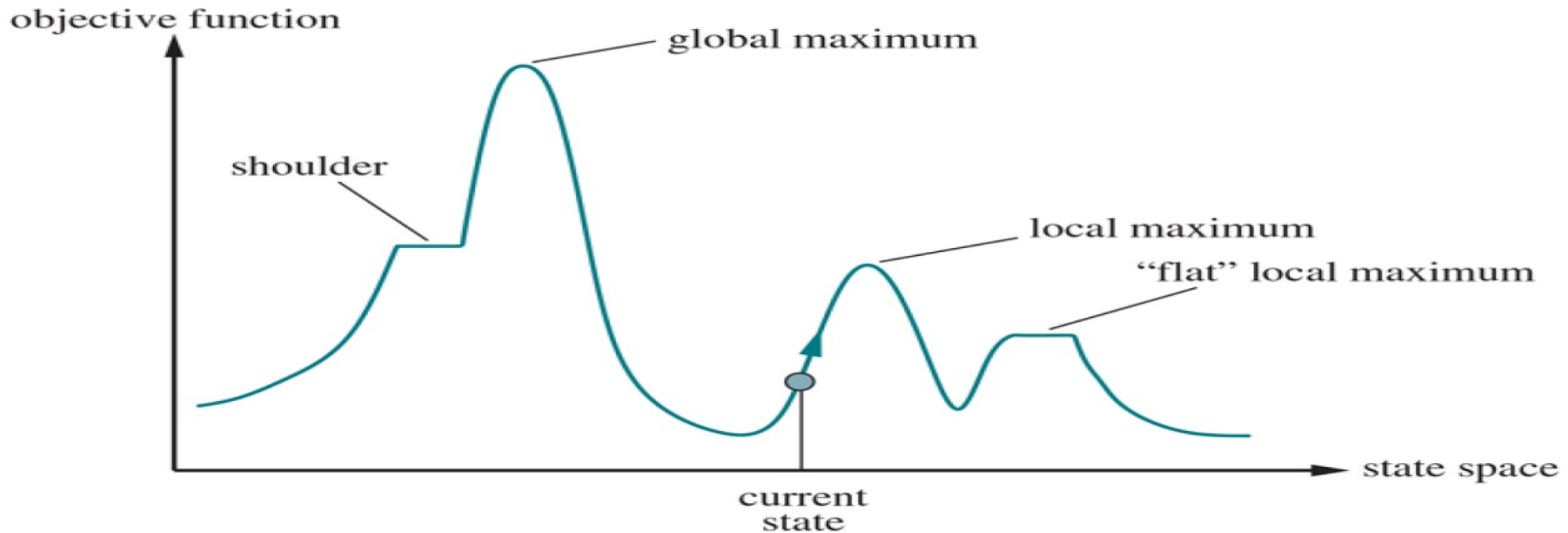
Local Search & Optimization Problem

- Pilih *state* awal dan mulai mencari solusi dari *state* tetangga terbaik
- Pemilihan *state* terbaik berdasarkan nilai fungsi tujuan (*objective function*)
- Algoritma untuk pencarian dengan optimasi:
 - Hill-Climbing
 - Simulated Annealing
 - Genetic Algorithm
 - Pemilihan *state* berdasarkan aturan seleksi alam yang diterapkan pada *state collection* (sering disebut sebagai populasi)



IF- Local Search & State Space Landscape

- Suatu *landscape* memiliki lokasi (*state*) dan *elevation* (*heuristic cost function* atau *objective function*)
 - Jika *elevation* \rightarrow *cost*, menemukan lembah terendah (global minimum)
 - Jika *elevation* \rightarrow *objective function*, menemukan puncak tertinggi (global maksimum)
 - *Local search* \rightarrow *complete*, akan selalu menemukan goal

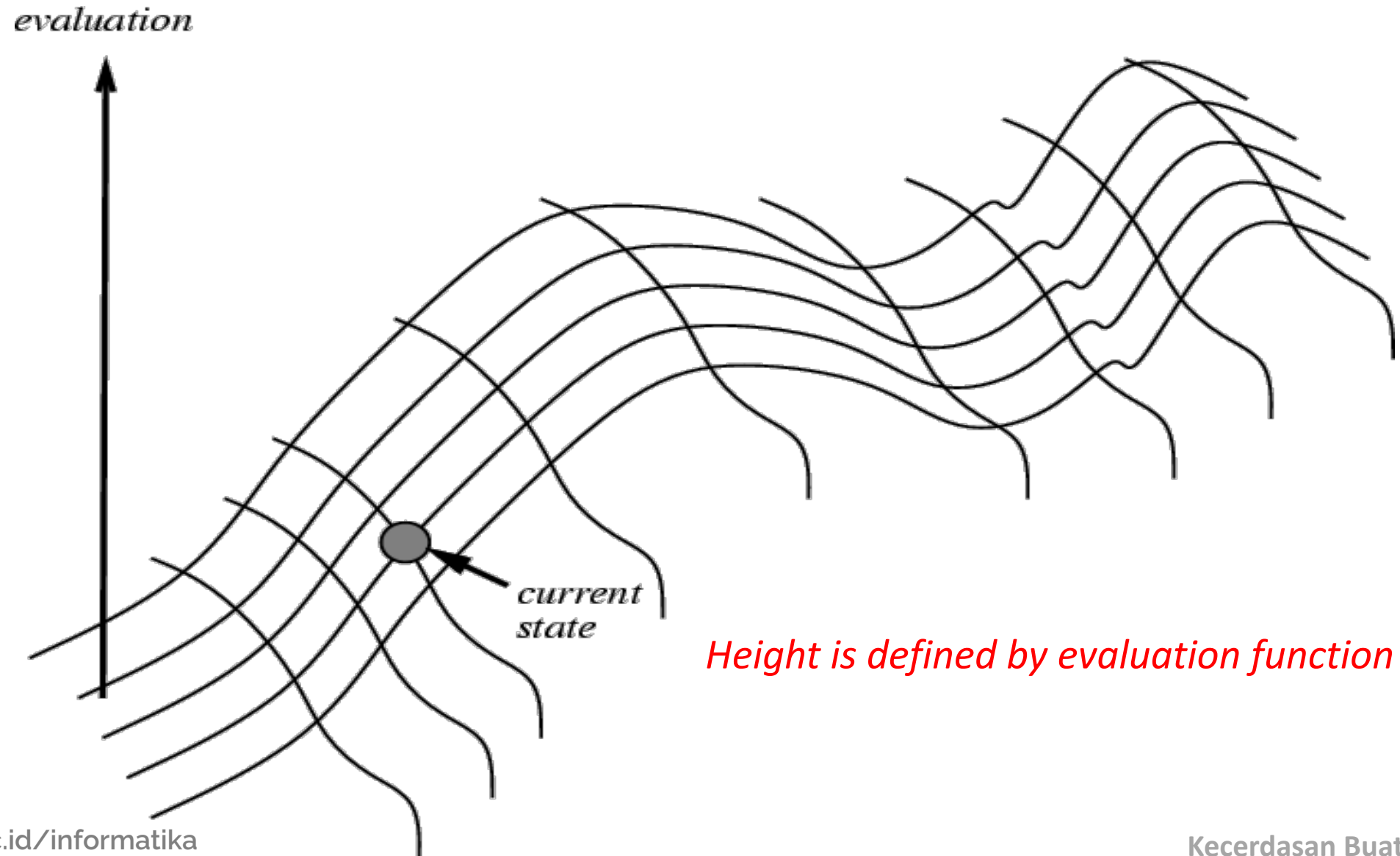


Hill Climbing Search

- Suatu iterasi yang secara kontinu bergerak ke arah peningkatan nilai → mendaki (*uphill*)
- Berhenti jika mencapai puncak (*peak*)
- Algoritma ini tidak menggunakan *search tree* sehingga struktur data untuk node digunakan hanya mencatat state dan nilai *objective function*

```
Function Hill-Climbing(problem) return a state that local maximum
input: problem // a problem
local variable : current // a node
                  neighbor // a node
current ← Make-Node(Initial State[problem])
while true do
    neighbor ← a highest value successor of current
    if Value[neighbor] <= Value[current] then return State[current]
    current ← neighbor
```


Hill climbing on a surface of states



Hill Climbing Algorithm

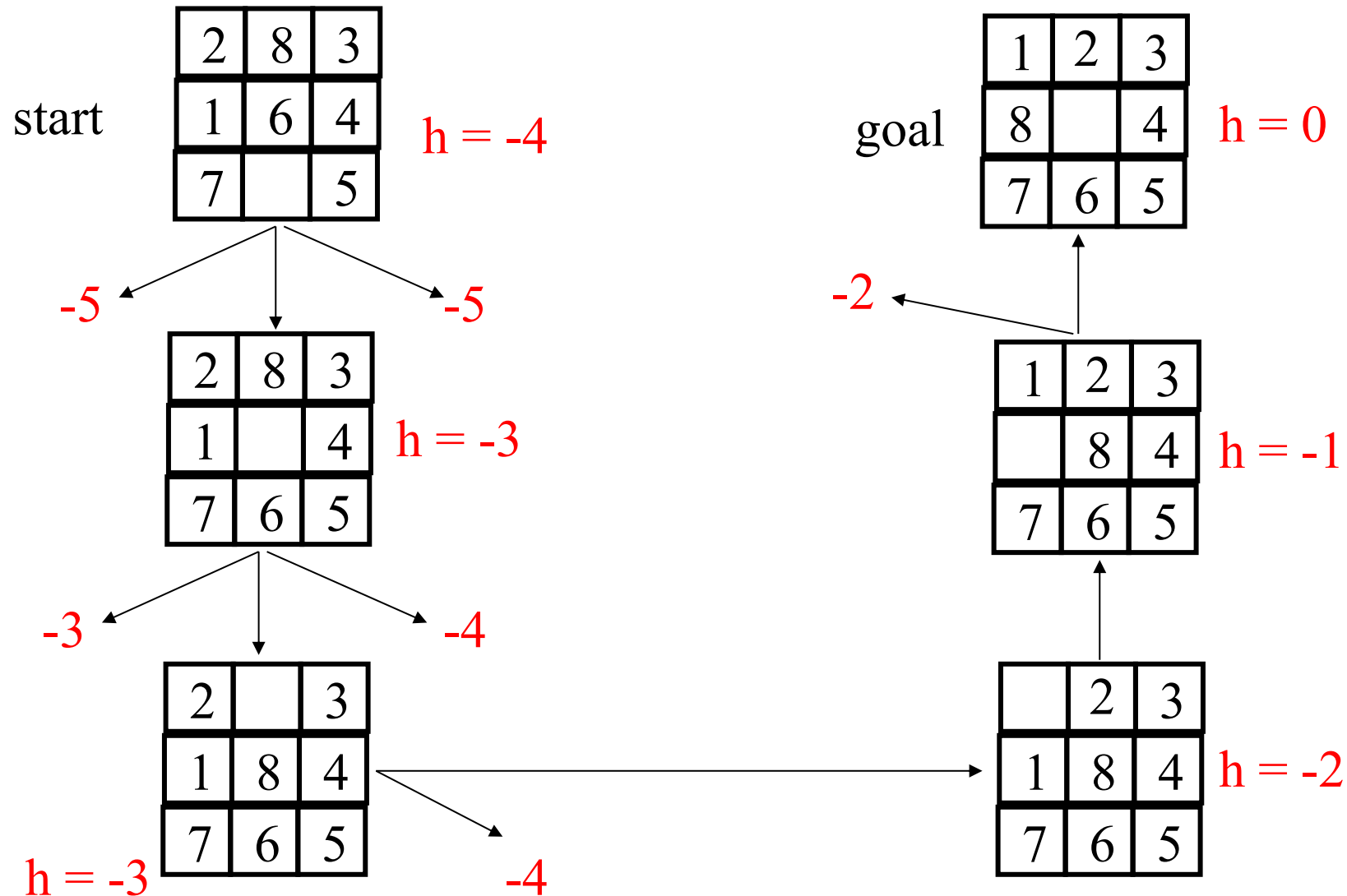
1. Pick a **random** point in the search space
2. Consider all the **neighbors** of the current state
3. Choose the neighbor with the **best quality** and move to that state
4. Repeat 2 - 4 until all the neighboring states are of lower quality
5. Return the current state as the **solution** state

Hill Climbing search disebut juga Greedy Local Search

➔ ambil state terdekat yang terlihat baik saat itu



IF- Contoh: Hill Climbing pada problem 8-puzzle



Exploring Landscape

- **Local Maxima:** puncak tapi bukan yang tertinggi di *state space*
- **Plateaus:** space dengan area datar yang membuat algoritma pencarian tidak punya arah (*random walk*)
- **Ridges:** datar seperti *plateau*, tetapi dengan *drop-offs* ke arah lain; dimana step ke arah utara, timur, dan barat mungkin turun, namun step ke arah barat laut mungkin naik.

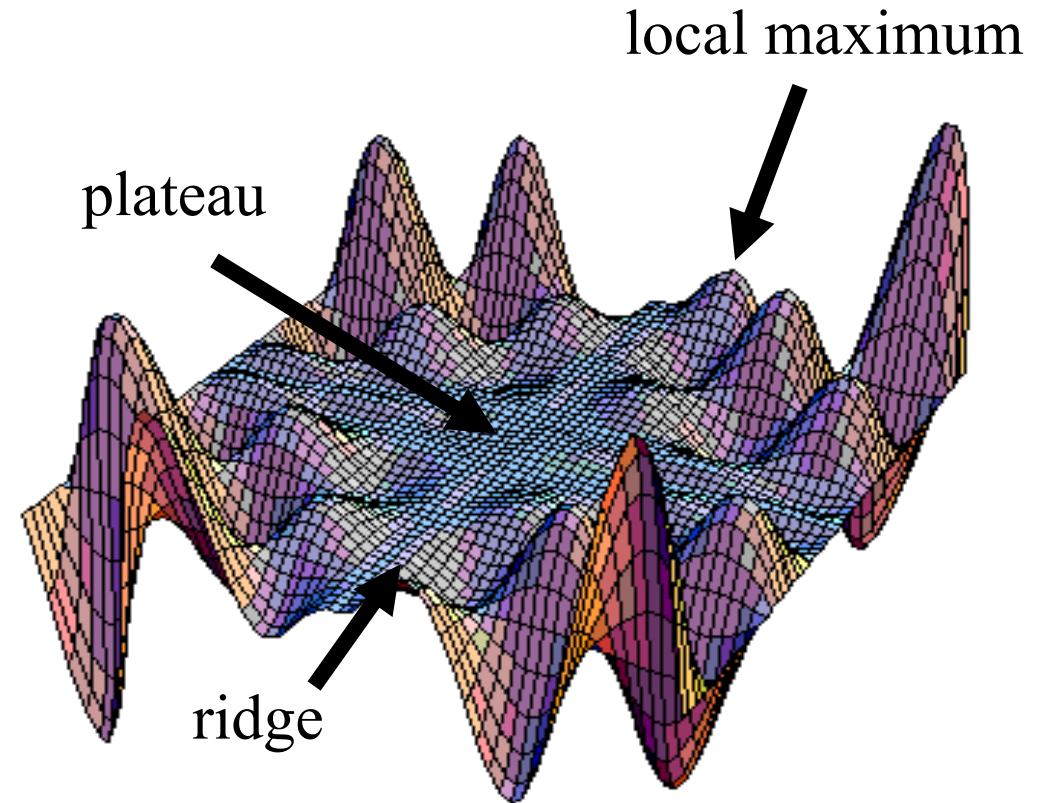


Image from: <http://classes.yale.edu/fractals/CA/GA/Fitness/Fitness.html>

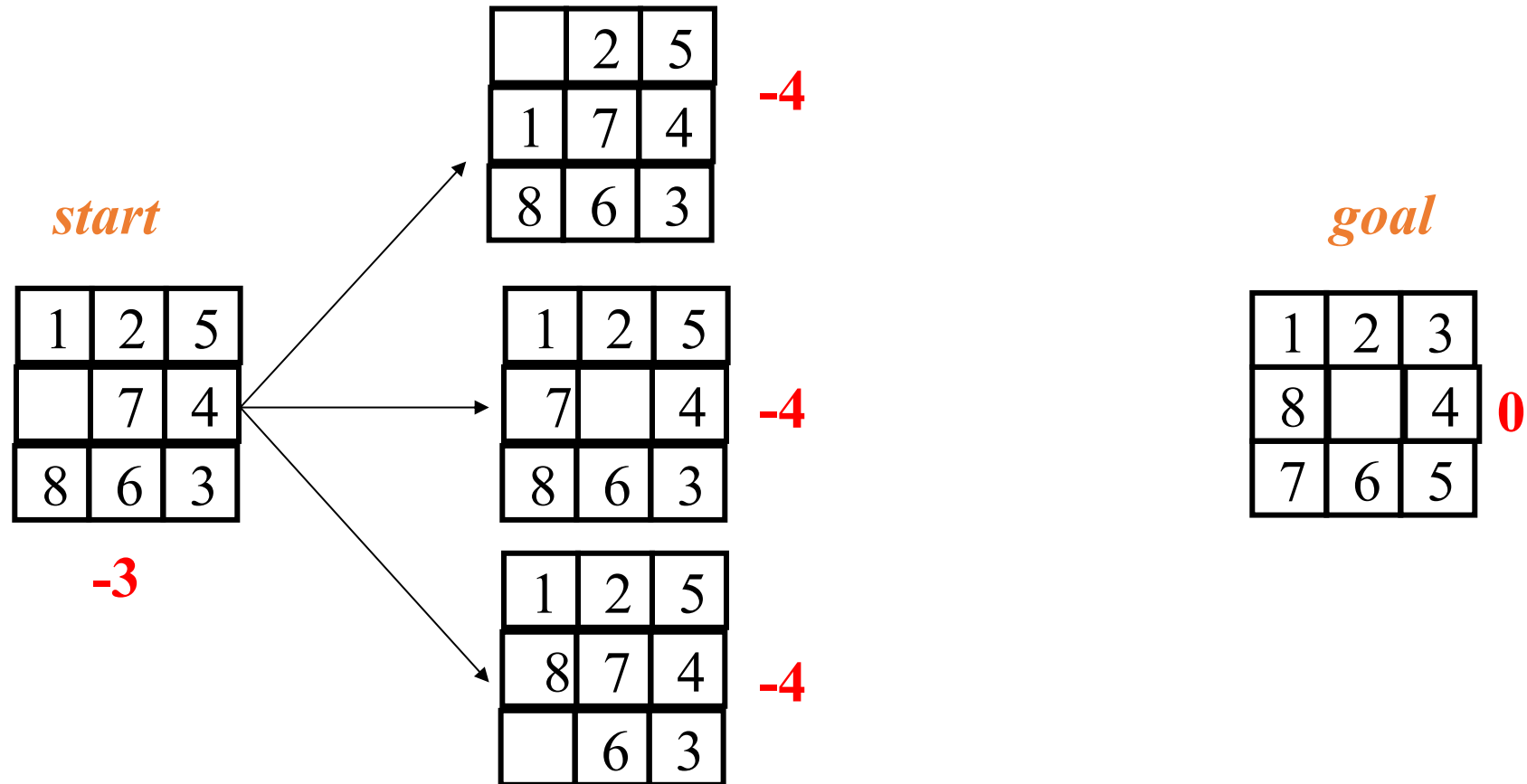


IF

Kelemahan Algoritma Hill Climbing

- Problem: *local maxima, plateaus, ridges*
- Solusi:
 - ***Random restart***: terus memulai kembali pencarian dari lokasi acak hingga tujuan ditemukan
 - ***Problem reformulation***: merumuskan kembali *search space* untuk mengeliminasi problem tersebut
- Beberapa problem *space* sangat bagus untuk hill climbing namun lainnya ada yang sangat buruk

Contoh: Kasus Local Optimum

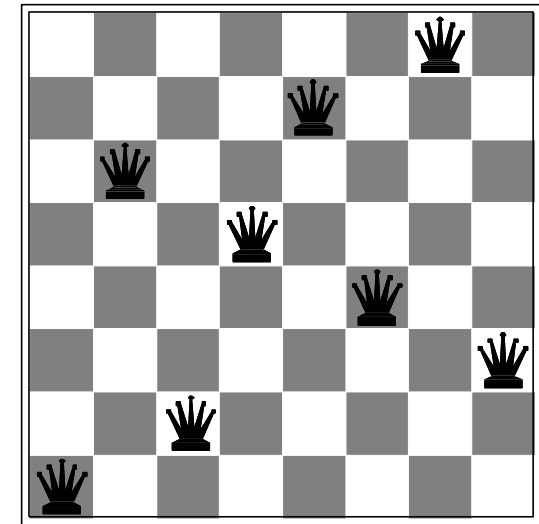




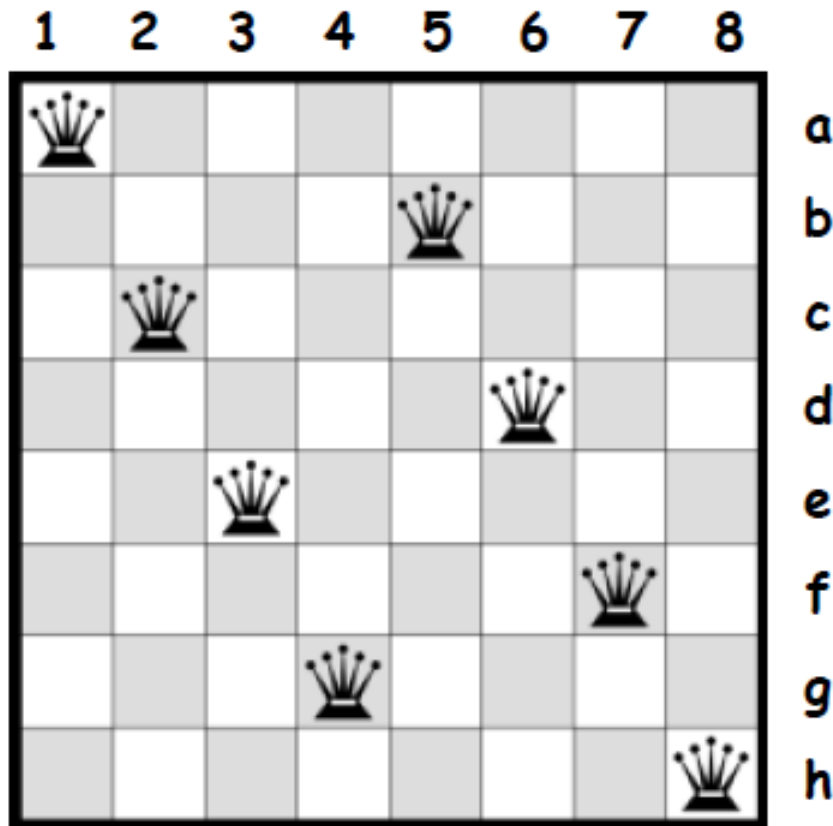
Tahapan Hill Climbing

(Contoh: problem 8-queens)

- Menentukan *initial state*
 - Secara random atau *state* dengan nilai objektif terkecil?
- Problem: 8-queens
 - Memilih *initial state*
 - Posisi bidak random dari 1-8 yang mempunyai nilai fungsi obyektif terkecil
 - Fungsi Obyektif
 - *Heuristic cost function* h = jumlah pasangan bidak ratu yang dapat saling menyerang



Contoh *Heuristic Cost Function* Problem 8-queens



$h = 1$ pasangan bidak ratu
yang dapat saling
menyerang

Yaitu 1a-8h



Contoh Hill Climbing untuk Problem 8-queens

Secara random, state current = 1e 2f 3g 4d 5e 6f 7g 8f

Nilai h (current) = 17 pasangan saling menyerang 1e-2f; 1e-3g; 1e-5e; 2f-3g; 2f-4d; 2f-6f; 2f-8f; 3g-5e; 3g-7g; 4d-5e; 4d-6f; 4d-7g; 5e-6f; 5e-7g; 6f-7g; 6f-8f; 7g-8f;

1	2	3	4	5	6	7	8	
18	12	14	13	13	12	14	14	a
14	16	13	15	12	14	12	16	b
14	12	18	13	15	12	14	14	c
15	14	14	♔	13	16	13	16	d
♔	14	17	15	♔	14	16	16	e
17	♔	16	18	15	♔	15	♔	f
18	14	♔	15	15	14	♔	16	g
14	14	13	17	12	14	12	18	h

- Hitung semua nilai h jika posisi satu bidak catur dirubah
- Misal 1:
posisi 1e dirubah ke 1d maka state 1d 2f 3g 4d 5e 6f 7g 8f; $h = 15$
- Misal 2:
posisi 2f dirubah ke 2e maka state 1e 2e 3g 4d 5e 6f 7g 8f; $h = 14$
- Nilai h terkecil adalah $h = 12$, jadi kondisi state dapat diubah ke:
 - 1e 2a 3g 4d 5e 6f 7g 8f
 -
 - 1e 2f 3g 4d 5e 6f 7h 8f (8 pilihan)

Aturan Hill Climbing, pilihan state selanjutnya nilai $h \leq 12$
 Misal current state → 1e 2a 3g 4d 5e 6f 7g 8f

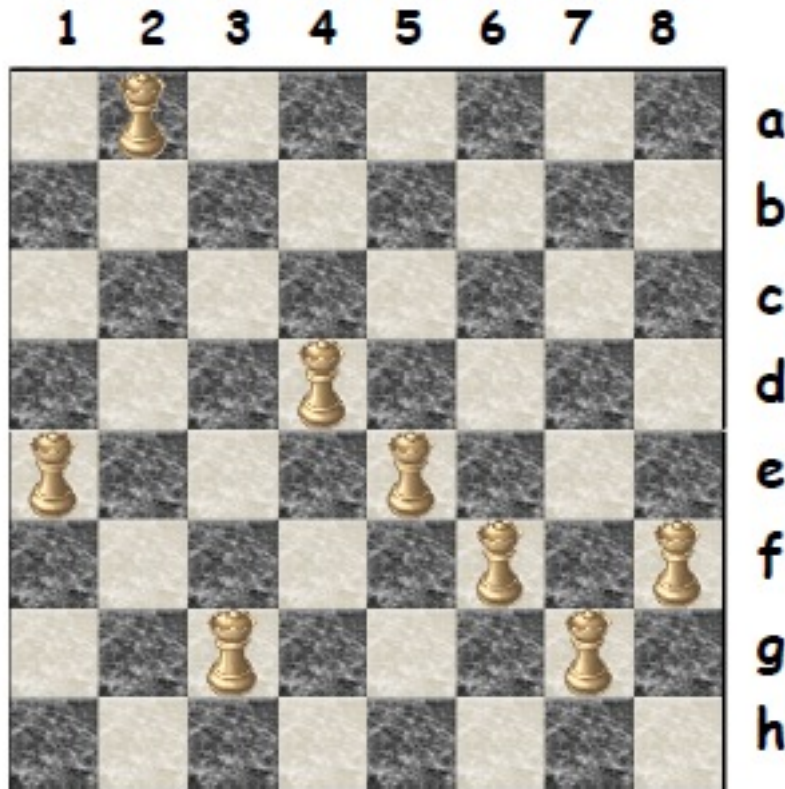


IF

Contoh Hill Climbing untuk Problem 8-queens

state current → 1e 2a 3g 4d 5e 6f 7g 8f

Nilai $h(\text{current}) = 12$ pasangan saling menyerang



- Hitung semua nilai h jika posisi satu bidak catur dirubah
- Nilai h terkecil adalah $h = ?$,
- Sehingga kondisi state dapat diubah ke:
... .. (? pilihan)
- Aturan Hill Climbing, pilihan *state* selanjutnya nilai $h \leq 12$

Proses berhenti saat nilai $h(\text{semua neighbor}) > h(\text{current})$ dan mengembalikan *state current* sebagai solusi

Contoh Hill Climbing untuk Mencari Jalur Terpendek

Pemasangan poster Gemastik dalam ITS akan dilakukan di Teknik Elektro(E), Teknik Informatika(C), Statistik (S), Matematika (M), Kantin (K). Jarak antar lokasi dalam kilometer (km) sebagai berikut:

	E	C	S	M	K
E	0	0.9	0.6	0.8	0.7
C		0	1.3	1.5	1.3
S			0	0.2	0.3
M				0	0.2
K					0

- Cari jalur pemasangan poster dengan jarak terpendek
- Nilai h = total jarak



Contoh Hill Climbing untuk Mencari Jalur Terpendek

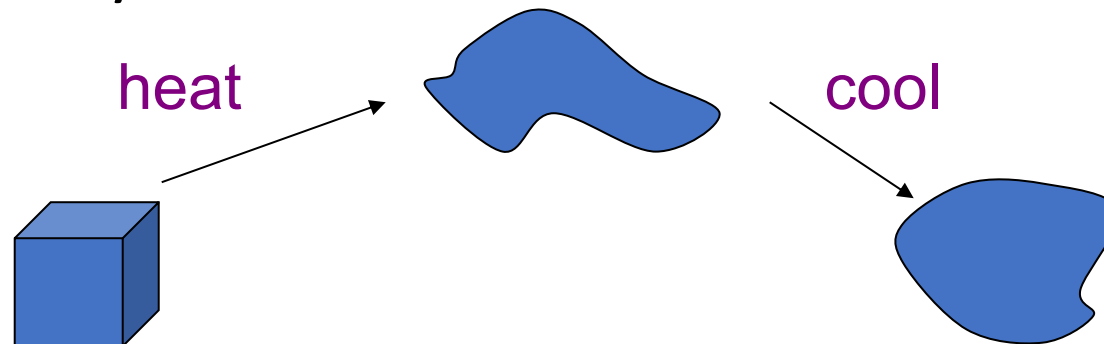
- Initial state = (E -> M -> C -> K -> S) secara random
 - Nilai $h = .8 + 1.5 + 1.3 + .3 = 3.9$
- Contoh proses swap: (A->B->C)
 - Tukar A-B didapat (B->A->C)
 - Tukar A-C didapat (C->B->A)
 - Tukar B-C didapat (A->C->B)
- Neighbor terbentuk dengan merubah (*swap*) 2 lokasi
 - (M -> E -> C -> K -> S) = 3.3 (C -> M -> E -> K -> S) = 3.3
 - (K-> M -> C -> E -> S) = 3.2 (S -> M -> C -> K -> E) = 3.7
 - (E -> C -> M -> K -> S) = 2.9 (E-> K -> C -> M-> S) = 3.7
 - (E -> S -> C -> K -> M) = 3.4 (E -> M -> K -> C -> S) = 3.6
 - (E -> M -> S -> K -> C) = 2.6 (E -> M -> C -> S -> K) = 3.9
- Hill-Climbing akan memilih $h = 2.6$
 - untuk current state (E -> M -> S -> K -> C)
- Proses swap dilakukan 1x lagi, dan solusi optimal didapat
 - (K->M->S->E->C) dengan $h = 1.9$

	E	C	S	M	K
E	0	0.9	0.6	0.8	0.7
C		0	1.3	1.5	1.3
S			0	0.2	0.3
M				0	0.2
K					0



Simulated Annealing

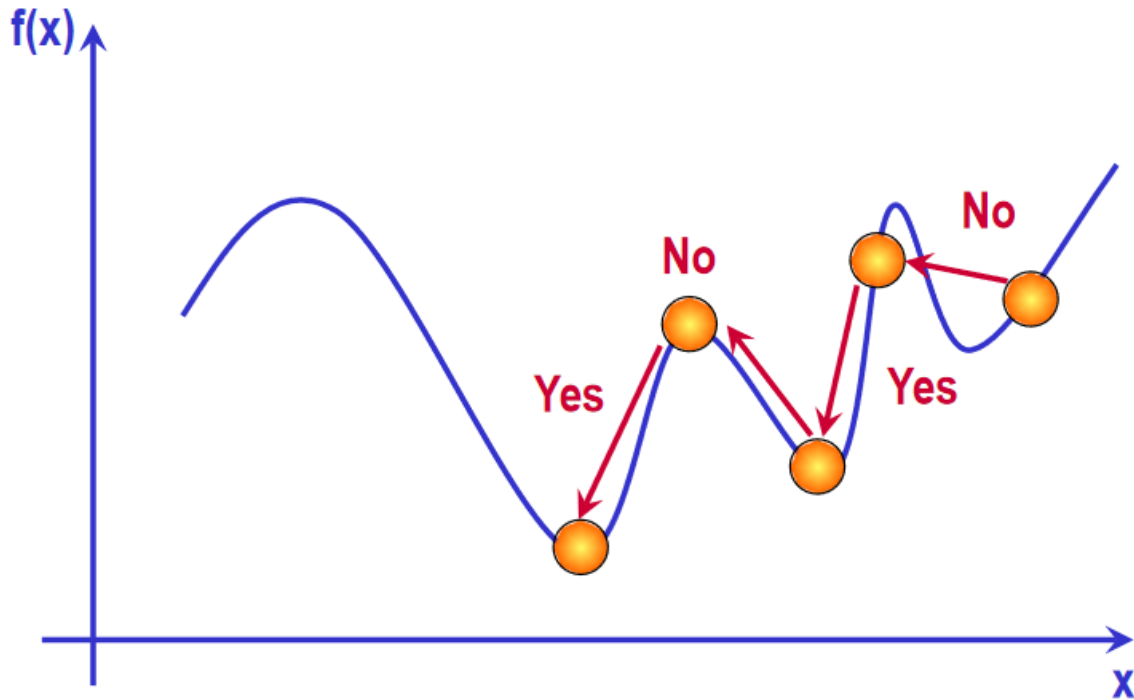
- **Ide:** keluar dari *local optimum* dengan melakukan “bad” move
- **Analogi:** ada bola yang menggelinding ke bawah (ke *local minima*), kemudian bidang tsb diguncang secukupnya agar bola bisa melambung sehingga tidak sampai ke *local minima*, tetapi tidak sampai keluar dari *global minima*.
- **Konsep Annealing** adalah proses penurunan suhu untuk memperoleh kondisi energi/suhu rendah pada sebuah material padat dari kondisi awal berupa cairan bersuhu sangat tinggi.
- **Konsep Simulated Annealing** : diawali dengan pemanasan dengan suhu yang sangat tinggi, kemudian secara berangsur-angsur menurunkan suhunya



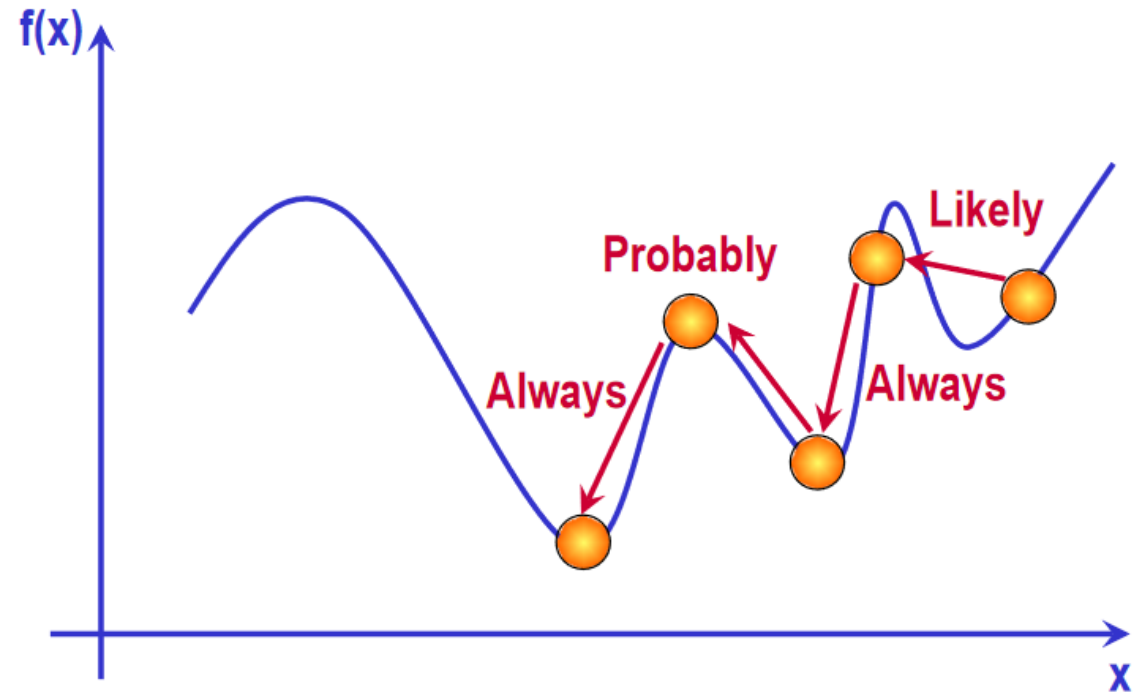
Simulated Annealing

- Untuk menerapkan *simulated annealing* dengan tujuan optimasi, diperlukan:
 - Fungsi *successor* yang mengembalikan solusi tetangga terdekat. Hal ini akan bekerja sebagai “*disturbance*” untuk partikel pada sistem.
 - Fungsi *target* untuk mengoptimasi yang tergantung pada *current state*. Fungsi ini akan bekerja sebagai energi pada sistem.
- Pencarian dimulai dengan *randomized state*. Pada setiap iterasi akan bergerak menuju state tetangga, selalu menerima pergerakan yang menurunkan energi dan hanya sekali menerima “*bad moves*” sesuai distribusi probabilitas yang tergantung “*temperature*” pada sistem.

Simulated Annealing



Local optimization attempts to reduce cost function at each iteration



Simulated annealing accept/reject new solution candidate based on probability

Sumber: https://users.ece.cmu.edu/~xinli/classes/cmu_18660/Lec28.pdf



Simulated Annealing Algorithm

Step 1: start from an initial point $X = X_0$ & $K = 0$

Step 2: evaluate cost function $F = f(X_K)$

Step 3: randomly move from X_K to a new solution X_{K+1}

Step 4: if $f(X_{K+1}) < F$, then

- ▼ Accept new solution
- ▼ $X = X_{K+1}$ & $F = f(X_{K+1})$

End if

Step 5: if $f(X_{K+1}) \geq F$, then

- ▼ Accept new solution with certain probability
- ▼ $X = X_{K+1}$ & $F = f(X_{K+1})$ iff $\text{rand}(1) < \varepsilon$

End if

Step 6: $K = K + 1$ & go to Step 2

Similar to local optimization

Help to get out of local minimum

Simulated Annealing Algorithm

Accept/reject new solution with the probability ε

- ▼ If $f(X_{K+1}) \geq F$, then
 - ▼ Accept new solution with certain **probability**
 - ▼ $X = X_{K+1}$ & $F = f(X_{K+1})$ iff **$\text{rand}(1) < \varepsilon$**
- ▼ End if

Use Boltzmann distribution to determine the probability ε

$$\varepsilon = \exp\left[-\frac{f(X_{K+1}) - F}{T_{K+1}}\right]$$

- ▼ T_{K+1} is a “temperature” parameter that gradually decreases
- ▼ E.g., $T_{K+1} = \alpha \cdot T_K$ where $\alpha < 1$



Contoh: Simulated Annealing untuk Travelling Salesman Problem (TSP)

Step 1: start from random route R , initial temperature T & $K = 1$

Step 2: evaluate cost function $F = f(R)$

Step 3: **define new route R_K by randomly swapping two cities**

Step 4: if $f(R_K) < F$, then

- ▼ Accept new route
- ▼ $R = R_K$ & $F = f(R_K)$

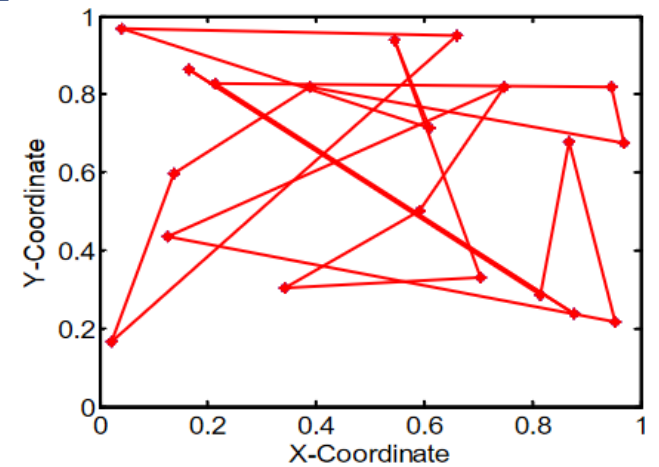
End if

Step 5: if $f(R_K) \geq F$, then

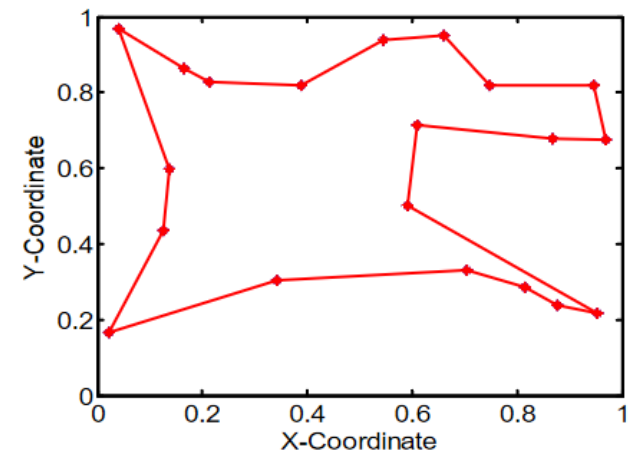
- ▼ Accept new solution with certain probability
- ▼ $R = R_K$ & $F = f(R_K)$ iff $\text{rand}(1) < \exp\{[F - f(R_K)]/T\}$

End if

Step 6: $T = \alpha T$ ($\alpha < 1$), $K = K + 1$, and go to Step 3



Initial route



Optimized route



- TERIMA KASIH -



Teknik Informatika
department of informatics
Fakultas Teknologi Informasi



www.its.ac.id



[its_campus](#)



[institut teknologi sepuluh nopember](#)