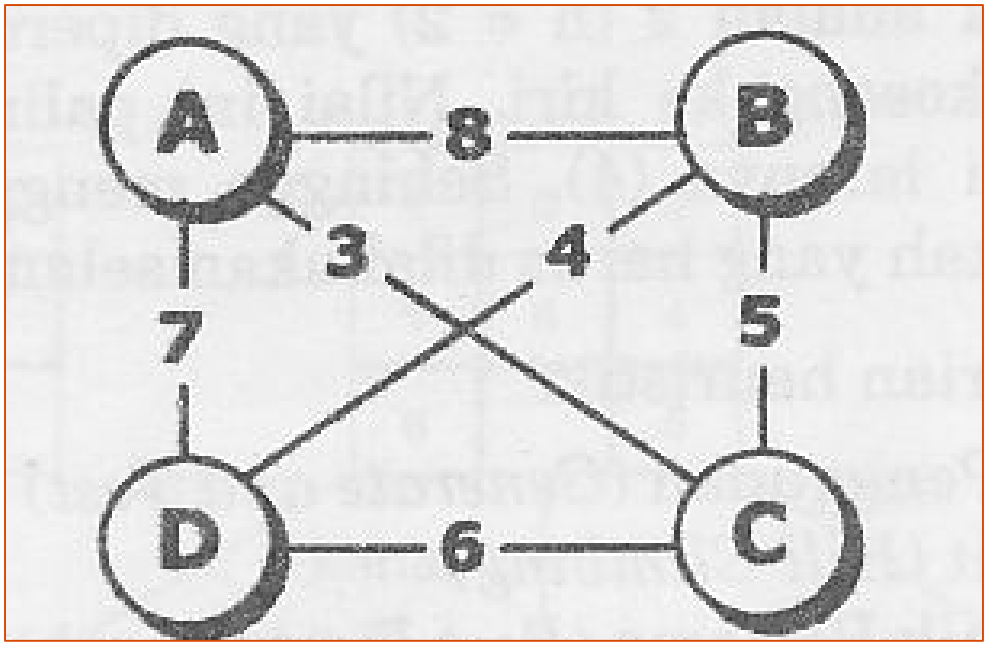
HEURISTIC SEARCH

Irvanizam Zamanhuri, M.Sc
Zulfan, S.Si, M.Sc
Dalila Husna Yunardi, B.Sc, M.Sc

Jurusan Informatika
Universitas Syiah Kuala

Travelling Salesmen Problem

- Seorang salesman ingin mengunjungi sejumlah n kota. Akan dicari rute terpendek di mana setiap kota hanya boleh dikunjungi tepat 1 kali.



Heuristic Search

- Pencarian buta tidak selalu dapat diterapkan dengan baik, hal ini disebabkan waktu aksesnya yang cukup lama serta besarnya memori yang dibutuhkan
- Kelemahan ini dapat diatasi jika ada informasi tambahan (fungsi heuristik) dari domain yang bersangkutan
- Heuristic adalah
  - Suatu proses yang mungkin dapat menyelesaikan suatu masalah tetapi tidak ada jaminan bahwa solusi yang dicari selalu dapat ditemukan
- Fungsi Heuristik adalah fungsi yang melakukan pemetaan (mapping) dari diskripsi keadaan masalah (problema) ke pengukur kebutuhan, umumnya direpresentasikan berupa angka.

Heuristic Search

- AI menggunakan heuristik dalam 2 situasi dasar :
  - Persoalan/problema yang mungkin memiliki solusi eksak, namun biaya perhitungan untuk menemukan solusi tersebut sangat tinggi dalam kebanyakan persoalan (seperti catur), ruang keadaan bertambah secara luar biasa seiring dengan jumlah
  - Persoalan yang mungkin tidak memiliki solusi eksak karena ambiquitas (ketidakpastian) mendasar dalam pernyataan persoalan atau data yang tersedia, diagnosa medis merupakan salah satu contohnya.
- Heuristik hanyalah sebuah cara menerka langkah berikutnya yang harus diambil dalam memecahkan suatu persoalan berdasarkan informasi yang ada/tersedia.

Metode Pencarian Heuristic

- Bangkitkan dan Uji (Generate and Test)
- HILL CLIMBING
  - Simple Hill Climbing
  - Steepest = Ascent Hill Climbing
- BEST FIRST SEARCH
  - Greedy Best First Search
  - Algoritma A*

Generate dan Test (1/2)

- Metode Generate--and--Test (GT) adalah metode yang paling sederhana dalam teknik pencarian heuristik.
- Didalam GT, terdapat dua prosedur penting:
  - Pembangkit (generate), yang membangkitkan semua solusi yang mungkin.
  - Test, yang menguji solusi yang dibangkitkan tersebut.
- Algoritma GT menggunakan prosedur Depth First Search karena suatu solusi harus dibangkitkan secara lengkap sebelum dilakukan Test.

Generate dan Test (2/2)

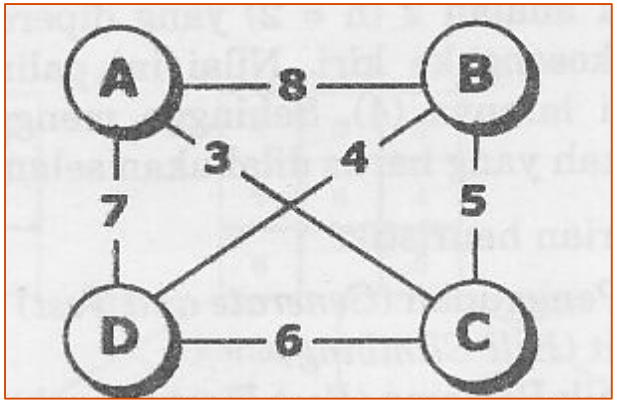
- Dengan penggunaan memori yang sedikit, DFS bisa digunakan sebagai prosedur pembangkit yang menghasilkan suatu solusi.
- Prosedur Test bisa menggunakan fungsi heuristik.

Algoritma Generate dan Test

1. Bangkitkan suatu kemungkinan solusi. Solusi bisa berupa suatu keadaan (state) tertentu. Solusi juga bisa berupa sebuah jalur dari satu posisi asal ke posisi tujuan, seperti dalam kasus pencarian rute dari satu kota asal ke kota tujuan.
2. Tes apakah solusi yang dibangkitkan tersebut adalah sebuah solusi yang bisa diterima sesuai dengan kriteria yang diberikan.
3. Jika solusi telah ditemukan, keluar. Jika belum, kembali ke langkah 1.

CONTOH KASUS TSP (Travelling Salesmen Problem)

- Seorang salesman ingin mengunjungi sejumlah n kota. Akan dicari rute terpendek di mana setiap kota hanya boleh dikunjungi tepat 1 kali.
- Jarak antara tiap–tiap kota sudah diketahui. Misalkan ada 4 kota dengan jarak antara tiap–tiap kota seperti terlihat pada gambar berikut.



CONTOH KASUS TSP(Travelling Salesmen Problem)

- Membangkitkan solusi--solusi yang mungkin dengan menyusun kota--kota dalam urutan abjad, yaitu :
  - A – B – C – D
  - A – B – D – C
  - A – C – B – D
  - A – C – D – B
  - Dst
- Untuk mengetahui jumlah seluruh kombinasi abjad yang mungkin menjadi sebuah solusi → **n!**

CONTOH KASUS TSP(Travelling Salesmen Problem)

- Pilihkeadaaan awal, misal ABCD dengan panjang lintasan 19
- Lakukan backtracking untuk mendapatkan lintasan ABDC, misal panjang lintasan 18
- Bandingkan lintasan ABDC dengan sebelumnya, lintasan terpendek akan dipilih untuk dilakukan backtracking lagi.
- Solusi terbaik adalah menemukan lintasan terpendek dari kota yang dilewati.

CONTOH KASUS
TSP(Travelling Salesmen Problem)

Percobaan ke-	Lintasan	Panjang Lintasan	Lintasan terpilih	Panjang Lintasan terpilih
1.	ABCD	19	ABCD	19
2.	ABDC	18	ABDC	18
3.	ACBD	12	ACBD	12
4.	ACDB	13	ACBD	12
5.	ADBC	16	ACBD	12
6.	ADCB	18	ACBD	12
7.	BACD	17	ACBD	12
8.	BADC	21	ACBD	12
9.	BCAD	15	ACBD	12
10.	BCDA	18	ACBD	12
11.	BDAC	14	ACBD	12
12.	BDCA	13	ACBD	12
13.	CABD	15	ACBD	12
14.	CADB	14	ACBD	12
15.	CBAD	20	ACBD	12
16.	CBDA	16	ACBD	12
17.	CDAB	21	ACBD	12

CONTOH KASUS
TSP(Travelling Salesmen Problem)

Lintasan	Panjang Lintasan	Lintasan terpilih
CDBA	18	ACBD
DABC	20	ACBD
DACB	15	ACBD
DBAC	15	ACBD
DBCA	12	ACBD atau DBCA
DCAB	17	ACBD atau DBCA
DCBA	19	ACBD atau DBCA

Generate dan Test

- Kelemahan :
  - Membangkitkan semua kemungkinan sebelum dilakukan pengujian
  - Membutuhkan waktu yang cukup besar dalam pencariannya

Hill Climbing

- Terdapat 2 jenis HC yang sedikit berbeda, yaitu Simple Hill Climbing (HC sederhana) dan Steepest--Ascent Hill Climbing (HC dengan memilih kemiringan yang paling tajam/curam).
- Simple HC, langsung memilih *new state* yang memiliki jalur yang lebih baik ("curam") daripada jalur--jalur sebelumnya tanpa memperhitungkan jalur--jalur lain yang lebih "curam".
- Sedangkan Steepest--Ascent HC, akan mengevaluasi semua state yang berada di bawah current state dan memilih state dengan jalur yang paling "curam".

Simple Hill Climbing

- Algoritma
  - Mulai dari keadaan awal, *(initial state)* lakukan pengujian,
    - jika state (keadaan) merupakan goal state (tujuan) →→berhenti
    - jika state (keadaan) bukan merupakan goal state (tujuan) →→lanjutkan dengan keadaan sekarang sebagai keadaan awal.
  - Ulangi langkah berikut hingga solusi ditemukan atau sampai tidak ada operator baru yang diaplikasikan pada keadaan sekarang
    - Pilih operator yang belum pernah digunakan
    - Gunakan operator untuk mendapatkan keadaan yang baru
  - Evaluasi keadaan baru tersebut :
    - Jika keadaan baru adalah tujuan →→keluar
    - Jika tidak tetapi nilainya lebih baik dari keadaan sekarang, →→jadikan keadaan baru tersebut menjadi keadaan sekarang
    - Jika keadaan baru tidak lebih baik dari keadaan sekarang →→lanjutkan iterasi

Simple Hill Climbing Contoh Kasus : TSP

- Operator yg digunakan adalah operator yang bisa menghasilkan kombinasi lintasan kota yang berbeda, yaitu dengan menukar urutan posisi 2 kota dalam suatu lintasan.
- Bila ada n kota maka kombinasi lintasan :

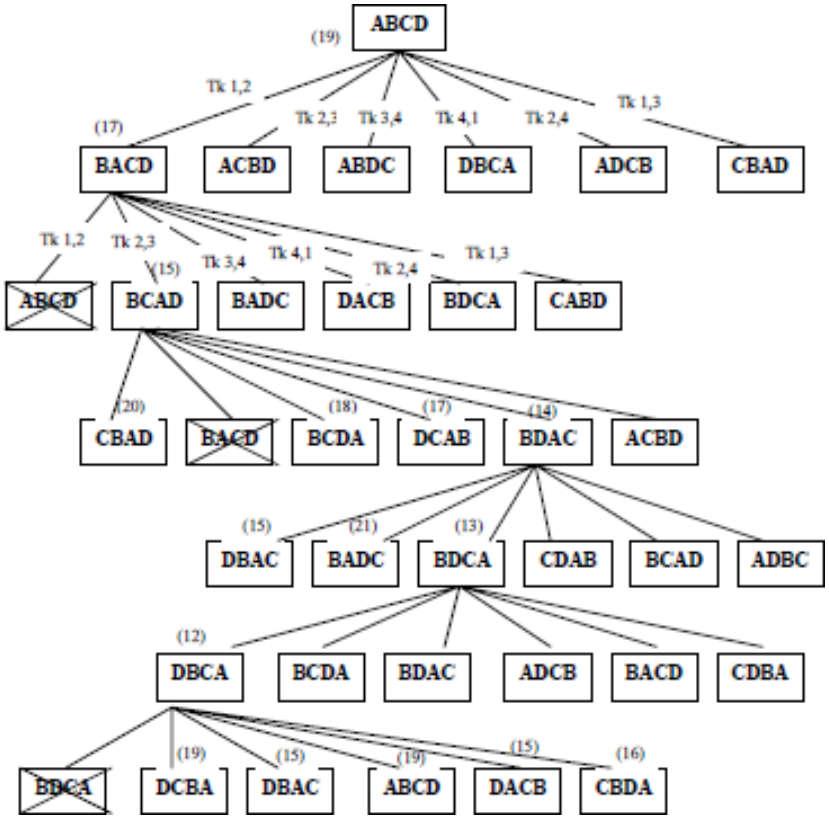
$$\frac{n!}{2!(n\ !\ 2)!}$$

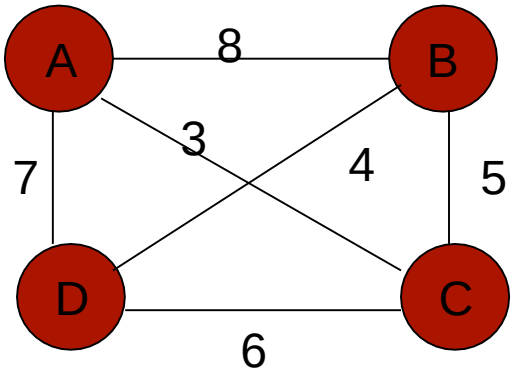
- Jika dari soal terdapat 4 kota maka kombinasi ada 6 yaitu :
  1. (1,2) tukar urutan posisi kota ke–1 dg kota ke–2
  2. (2,3) tukar urutan posisi kota ke–2 dg kota ke–3
  3. (3,4) tukar urutan posisi kota ke–3 dg kota ke–4
  4. (4,1) tukar urutan posisi kota ke–4 dg kota ke–1
  5. (2,4) tukar urutan posisi kota ke–2 dg kota ke–4
  6. (1,3) tukar urutan posisi kota ke–1 dg kota ke–3

Simple Hill Climbing Contoh Kasus : TSP

- Padapencarian ini, penggunaan urutan dari kombinasi harus konsisten. Setelah kombinasi ditentukan, gunakan algoritma pengerjaan sesuai aturan *metode simple hill climbing*. Misalnya keadaan awal adalah ABCD

Simple Hill Climbing
Contoh Kasus : TSP dengan 6 operator





Penjelasan (1/2)

- Keadaan awal, lintasan ABCD (=19).
- Level pertama, hill climbing mengunjungi BACD (=17), BACD (=17) < ABCD (=19), sehingga
  - BACD menjadi pilihan selanjutnya dengan operator Tukar 1,2
- Level kedua, mengunjungi ABCD, karena operator Tukar 1,2 sudah dipakai BACD, maka pilih node
  - lain yaitu BCAD (=15), BCAD (=15) < BACD (=17)
- Level ketiga, mengunjungi CBAD (=20), CBAD (=20) > BCAD (=15), maka pilih node lain yaitu
    - BCDA (=18), pilih node lain yaitu DCAB (=17), pilih node lain yaitu BDAC (=14), BDAC (=14) < BCAD (=15)

Penjelasan (2/2)

- Level keempat, mengunjungi DBAC (=15), DBAC(=15) > BDAC (=14), maka pilih node lain yaitu
  - BADC (=21), pilih node lain yaitu BDCA (=13), BDCA (=13) < BDAC (=14)
- Level kelima, mengunjungi DBCA (=12), DBCA (=12) < BDCA (=13)
- Level keenam, mengunjungi BDCA, karena operator Tukar 1,2 sudah dipakai DBCA, maka pilih node
  - lain yaitu DCBA, pilih DBAC, pilih ABCD, pilih DACB, pilih CBDA
- Karena sudah tidak ada node yang memiliki nilai heuristik yang lebih kecil dibanding nilai heuristik DBCA, maka **node DBCA** (=12) adalah **lintasan terpendek (SOLUSI)**

Simple Hill Climbing
Contoh Kasus : TSP dengan 6 operator

- Pencarian dilihat dari anak kiri, bila nilai heuristik anak kiri lebih baik, maka dibuka untuk pencarian selanjutnya, bila tidak baru melihat tetangga dari anak kiri tersebut.
- Solusi yang dihasilkan adalah node DBCA (=12) →→ lintasan terpendek dibanding yang lain.
- Kelemahannya :

  1. tidak semua solusi dapat ditemukan seperti

     pada metode *generate and test (2 solusi)*.

  2. pembatasan kombinasi operator →→

     penemuan solusi yang tidak maksimal

Simple Hill Climbing

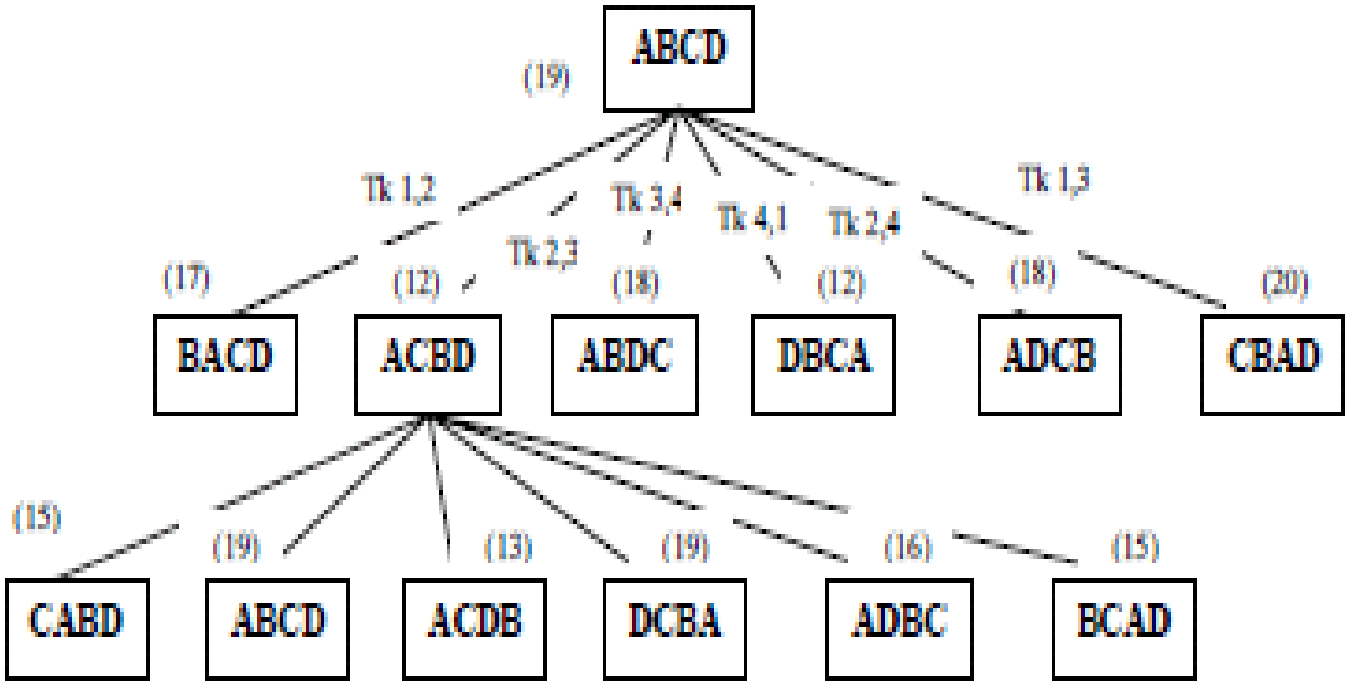
- Masalah--masalah yang mungkin timbul pada prosedur Hill Climbing :
  - Maksimum Lokal
    - Suatu keadaaan yang lebih baik daripada semua tetangganya namum masih belum lebih baik dari suatu keadaan lain yang jauh letaknya darinya.
  - Daratan (Plateau)
    - Suatu daerah datar dari ruang pencarian (search) dimana semua himpunan keadaan tetangganya memiliki nilai yang sama.
  - Punggung (Ridge)
    - Suatu daerah ruang pencarian (search) yang lebih tinggi daripada daerah sekitarnya, namun tidak dapat dibalikkan oleh langkah--langkah tunggal ke arah manapun.

Simple Hill Climbing

- Solusinya :
  - Melakukan langkah balik (tracking) ke simpul yang lebih awal dan mencoba bergerak ke arah yang lain.
  - Melakukan lompatan besar ke suatu arah untuk mencoba bagian ruang pencarian yang baru.
  - Menerapkan dua atau lebih aturan sebelum melakukan uji coba. Ini bersesuaian dengan bergerak kebeberapa arah sekaligus.

STEEPEST – ASCENT HILL CLIMBING

- Hampirsama dengan simple hill climbing, hanya saja gerakan pencarian tidak dimulai dari kiri, tetapi berdasarkan nilai heuristik terbaik.


ABCD
(19)
Tk 1,2
Tk 2,3
Tk 3,4
Tk 4,1
Tk 2,4
Tk 1,3
(17)
(12)
(18)
(12)
(18)
(20)
BACD
ACBD
ABDC
DBCA
ADCB
CBAD
(15)
(19)
(13)
(19)
(16)
(15)
CABD
ABCD
ACDB
DCBA
ADBC
BCAD

Keadaan awal, lintasan ABCD (=19).
Level pertama, hill climbing memilih nilai heuristik terbaik yaitu ACBD (=12) sehingga ACBD menjadi pilihan selanjutnya.

Level kedua, hill climbing memilih nilai heuristik terbaik, karena nilai heuristik lebih besar dibanding ACBD, maka hasil yang diperoleh lintasannya tetap ACBD (=12)