

KECERDASAN BUATAN

Simple Hill Climbing



Disusun Oleh:

1. Lutvi Maulida Al H. (081112006)
2. Nurul Fauziah (081112021)
3. Anggraeni Susanti (081112055)
4. Syahrul Bahar Hamdani (081211232012)

Departemen Matematika
Fakultas Sains dan Teknologi
Universitas Airlangga

2014

DAFTAR ISI

Halaman Judul	i
Daftar Isi	ii
Bab I Pendahuluan	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Tujuan	2
Bab II Pembahasan	3
2.1 Pengertian <i>Hill Climbing</i>	3
2.2 Algoritma <i>Simple Hill Climbing</i>	3
2.3 Contoh Penerapan Algoritma <i>Simple Hill Climbing</i>	4
Bab III Penutup	9
3.1 Simpulan	9
Daftar Pustaka	10

BAB I

PENDAHULUAN

1.1. Latar Belakang

Heuristic seringkali disebut sebagai lawan dari kata algoritmik dalam dunia pemrograman. *Heuristic* adalah suatu proses yang mungkin dapat menyelesaikan suatu masalah tetapi tidak ada jaminan bahwa solusi yang dicari selalu dapat ditemukan. Dengan kata lain, *heuristic* adalah sebuah teknik yang mengembangkan efisiensi dalam proses pencarian, namun dengan kemungkinan mengorbankan kelengkapan (*completeness*).

Sedangkan fungsi heuristik adalah fungsi yang melakukan pemetaan (mapping) dari diskripsi keadaan masalah (problema) ke pengukur kebutuhan, umumnya dipresentasikan berupa angka. Fungsi heuristik digunakan untuk mengevaluasi keadaan-keadaan problema individual dan menentukan seberapa jauh hal tersebut dapat digunakan untuk mendapatkan solusi yang diinginkan.

Dalam mempelajari metode-metode pencarian heuristik, kata heuristik diartikan sebagai suatu fungsi yang memberikan suatu nilai berupa biaya perkiraan (estimasi) dari suatu solusi. Ada beberapa macam metode pencarian heuristik, yaitu pembangkitan dan pengujian (*Generate and Test*), *Hill Climbing*, *Best First Search*, *Alpha Beta Prunning*, *Means-End-Anlysis*, *Constraint Satisfaction*, dan *Simulated Annealing*.

Dalam metode pencarian heuristik *Hill Climbing*, ada dua macam metode heuristik yakni *Simple Hill Climbing* dan *Steepest (Ascent Hill Climbing)*. Dalam makalah ini, penulis akan membahas mengenai metode heuristik *Simple Hill Climbing* yang disertai dengan contoh algoritma pada metode tersebut pada saat diterapkan pada suatu permasalahan.

1.2. Rumusan Masalah

Rumusan masalah yang akan dibahas dalam makalah ini adalah sebagai berikut;

1.2.1. Apakah pengertian *Hill Climbing*?

1.2.2. Bagaimanakah algoritma *Simple Hill Climbing*?

1.2.3. Bagaimanakah contoh penerapan algoritma *Simple Hill Climbing*?

1.3. Tujuan

Tujuan dari makalah ini adalah sebagai berikut;

1.3.1. Mengetahui pengertian *Hill Climbing*.

1.3.2. Mengetahui algoritma *Simple Hill Climbing*.

1.3.3. Mengetahui contoh dan penerapan algoritma *Simple Hill Climbing*.

BAB II PEMBAHASAN

2.1 Pengertian *Hill Climbing*

Metode *Hill Climbing* hampir sama dengan metode pembangkitan & pengujian (*Generate and Test*), hanya saja proses pengujian dilakukan dengan menggunakan fungsi heuristik. Pembangkitan keadaan berikutnya sangat tergantung pada *feedback* dari prosedur pengetesan. Tes yang berupa fungsi heuristik ini akan menunjukkan seberapa baiknya nilai terkaan yang diambil terhadap keadaan-keadaan lainnya yang mungkin.

Hill Climbing adalah proses pengujian yang dilakukan dengan menggunakan fungsi heuristik. Pembangkitan keadaan berikutnya sangat tergantung pada *feedback* dari prosedur pengetesan. Tes yang berupa fungsi heuristik ini akan menunjukkan seberapa baiknya nilai terkaan yang diambil terhadap keadaan-keadaan lainnya yang mungkin.

Metode *Hill climbing* merupakan variasi dari *depth-first search*. Dengan metode ini, eksplorasi terhadap keputusan dilakukan dengan cara *depth-first search* dengan mencari *path* yang bertujuan menurunkan *cost* untuk menuju kepada *goal*/keputusan. Yaitu dengan selalu memilih nilai heuristik terkecil.

Dalam metode heuristik *Hill Climbing*, terdapat dua jenis *Hill Climbing* yang sedikit berbeda, yakni *Simple Hill Climbing* (*Hill Climbing* sederhana) dan *Steepest-Ascent Hill Climbing* (*Hill Climbing* dengan memilih kemiringan yang paling tajam / curam).

2.2 Algoritma *Simple Hill Climbing*

Berikut adalah algoritma dari *Simple Hill Climbing*;

1. Mulai dari keadaan awal, lakukan pengujian: jika merupakan tujuan, maka berhenti; dan jika tidak, lanjutkan dengan keadaan sekarang sebagai keadaan awal.

2. Kerjakan langkah-langkah berikut sampai solusinya ditemukan, atau sampai tidak ada operator baru yang akan diaplikasikan pada keadaan sekarang:
 - a) Cari operator yang belum pernah digunakan; gunakan operator ini untuk mendapatkan keadaan yang baru.
 - b) Evaluasi keadaan baru tersebut.
 - 1) Jika keadaan baru merupakan tujuan, keluar.
 - 2) Jika bukan tujuan, namun nilainya lebih baik daripada keadaan sekarang, maka jadikan keadaan baru tersebut menjadi keadaan sekarang.
 - 3) Jika keadaan baru tidak lebih baik daripada keadaan sekarang, maka lanjutkan iterasi.

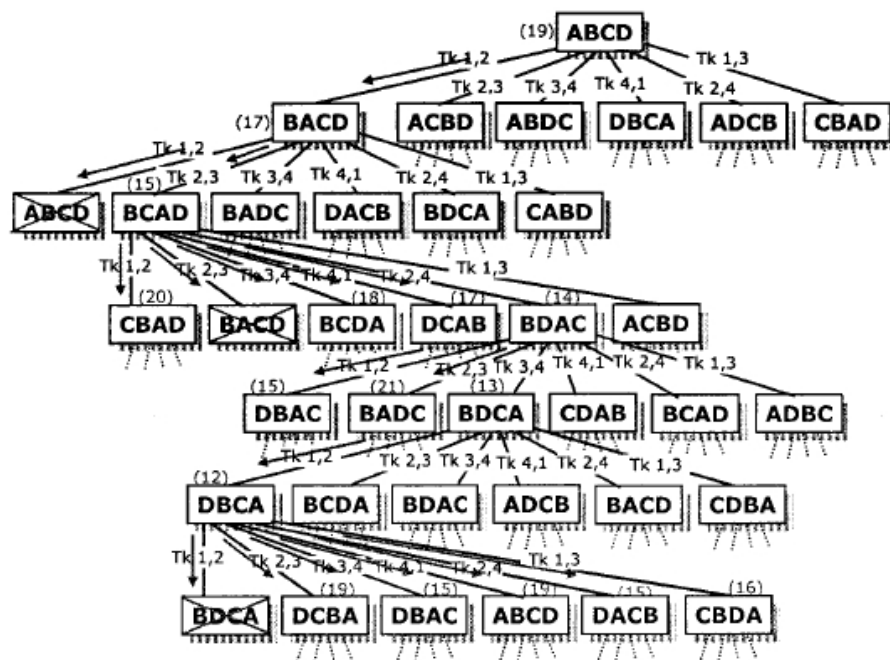
Pada *simple hill climbing* ini, ada 3 masalah yang mungkin, yaitu:

- 1) Algoritma akan berhenti kalau mencapai nilai optimum lokal.
- 2) Urutan penggunaan operator akan sangat berpengaruh pada penemuan solusi.
- 3) Tidak diijinkan untuk melihat satupun langkah sebelumnya.

2.3 Contoh Penerapan Algoritma *Simple Hill Climbing*

Salah satu contoh dari penerapan Algoritma *Simple Hill Climbing* adalah Traveling Salesman Problem.

Disini ruang keadaan berisi semua kemungkinan lintasan yang mungkin. Operator digunakan untuk menukar posisi kota-kota yang bersebelahan. Fungsi heuristik yang digunakan adalah panjang lintasan yang terjadi.



Gambar 2.22 Metode Simple Hill Climbing dengan 6 operator.

Operator yang akan kita gunakan, adalah menukar urutan posisi 2 kota dalam suatu lintasan. Apabila ada n kota, dan kita ingin mencari kombinasi lintasan dengan menukar posisi urutan 2 kota, maka kita akan mendapatkan sebanyak :

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

Sehingga kalau ada 4 kota, kita bisa memperoleh : $\frac{4!}{2 \times (4-2)!} = 6$ kombinasi.

Keenam kombinasi ini akan kita pakai semuanya sebagai operator, yaitu:

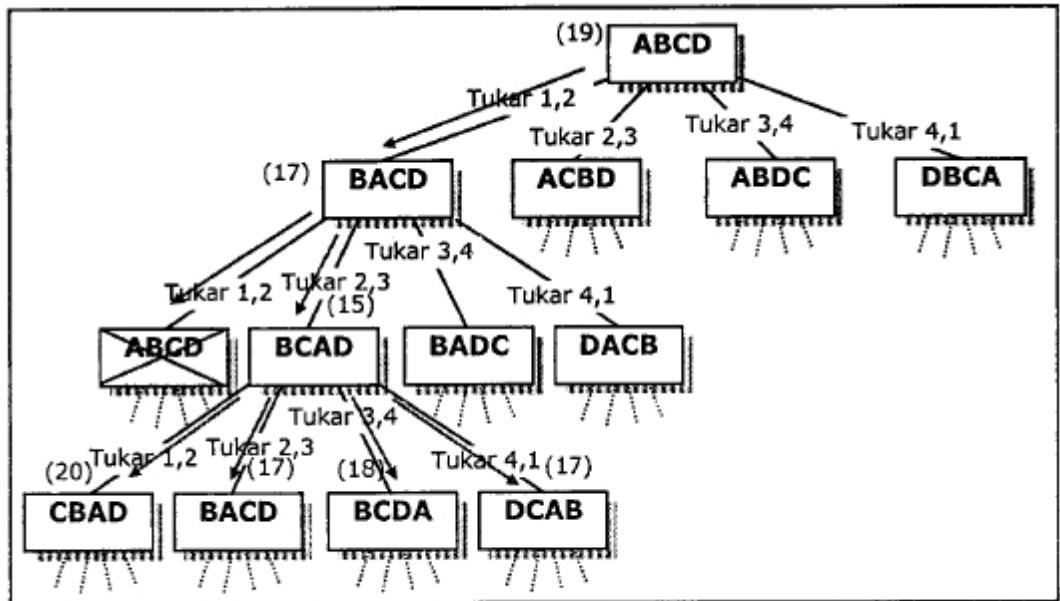
- * Tukar 1, 2 (menukar urutan posisi kota ke-1 dengan kota ke-2).
- * Tukar 2, 3 (menukar urutan posisi kota ke-2 dengan kota ke-3).
- * Tukar 3, 4 (menukar urutan posisi kota ke-3 dengan kota ke-4).
- * Tukar 4, 1 (menukar urutan posisi kota ke-4 dengan kota ke-1).
- * Tukar 2, 4 (menukar urutan posisi kota ke-2 dengan kota ke-4).
- * Tukar 1, 3 (menukar urutan posisi kota ke-1 dengan kota ke-3).

Pada Gambar 2.22 terlihat bahwa, pada keadaan awal, lintasan terpilih adalah ABCD (=19). Pada level pertama, *hill climbing* akan mengunjungi BACD (=17) yang ternyata memiliki nilai heuristik lebih kecil dibandingkan dengan ABCD ($17 < 19$), sehingga BACD menjadi pilihan selanjutnya dengan operator terpakai Tukar1,2. Pada level kedua, *hill climbing* akan mengunjungi ABCD. Karena operator Tukar 1, 2 sudah digunakan oleh BACD, maka

dipilih *node* yang lain yaitu BCAD (=15). Karena nilai heuristik BCAD lebih kecil dibanding dengan BACD ($15 < 17$), maka *node* BCAD akan menjadi pilihan selanjutnya dengan operator Tukar_{2,3}. Kemudian *hill climbing* akan mengunjungi CBAD (=20). Karena nilai heuristik CBAD lebih besar jika dibanding dengan BCAD ($20 > 17$), maka dipilih *node* lain. Pencarian menuju ke *node* BACD, karena operator Tukar_{2,3} sudah pernah digunakan oleh BCAD, maka dipilih *node* lain. Kunjungan berikutnya ke *node* BCDA (=18). Nilai inipun masih lebih besar dari nilai heuristik BCAD, sehingga dipilih *node* lain. *Node* yang dikunjungi berikutnya adalah DCAB (=19). Nilai heuristik DCAB ternyata juga lebih besar dibanding dengan BCAD, sehingga pencarian dilanjutkan di *node* lainnya lagi, yaitu BDAC (=14). Nilai heuristik ini sudah lebih kecil daripada nilai heuristik *node* BCAD ($14 < 15$), maka sekarang *node* ini yang akan dieksplorasi. Pencarian pertama ditemukan *node* DBAC (=21), yang lebih besar daripada nilai BDAC. Nilai heuristik yang lebih kecil diperoleh pada *node* BDCA (=13). Sehingga *node* BDCA ini akan dieksplorasi. Pencarian pertama sudah mendapatkan *node* dengan nilai heuristik yang lebih kecil, yaitu DBCA (=12). Sehingga *node* ini dieksplorasi juga. Dari hasil eksplorasi dengan pemakaian semua operator, ternyata sudah tidak ada *node* yang memiliki nilai heuristik yang lebih kecil dibanding dengan nilai heuristik DBCA, sehingga sebenarnya *node* DBCA (=12) inilah lintasan terpendek yang kita cari (SOLUSI). Misalkan kita tidak menggunakan semua operator, melainkan kita hanya menggunakan 4 operator pertama saja, yaitu :

- * Tukar 1,2 (menukar urutan posisi kota ke'1 dengan kota ke'2).
- * Tukar 2, 3 (menukar urutan posisi kota ke-2 dengan kota ke'3).
- * Tukar 3,4 (menukar urutan posisi kota ke-3 dengan kota ke'4).
- * Tukar 4, 1 (menukar urutan posisi kota ke-4 dengan kota ke'1).

Maka pencarian dengan *simple hill climbing* ini dapat dilihat pada Gambar 2.23. Lintasan terpendek yang diperoleh adalah B-C-A-D yaitu sebesar 15. Disini kita akan terjebak pada nilai minimum local yang disebabkan oleh kurangnya operator yang kita gunakan. Kita tidak dapat memperoleh nilai minimum globalnya yaitu sebesar 12.



Gambar 2.23. Metode Simple Hill Climbing dengan 4 operator.

Contoh TSP di atas adalah contoh pertama yang dibahas dalam makalah ini. Untuk contoh kedua adalah *game Number Puzzle Slider* dengan langkah-langkah menggunakan Simple Hill Climbing.

Berikut adalah contoh menyelesaikan *Number Puzzle Slider* dengan kondisi tertentu menggunakan Simple Hill Climbing;

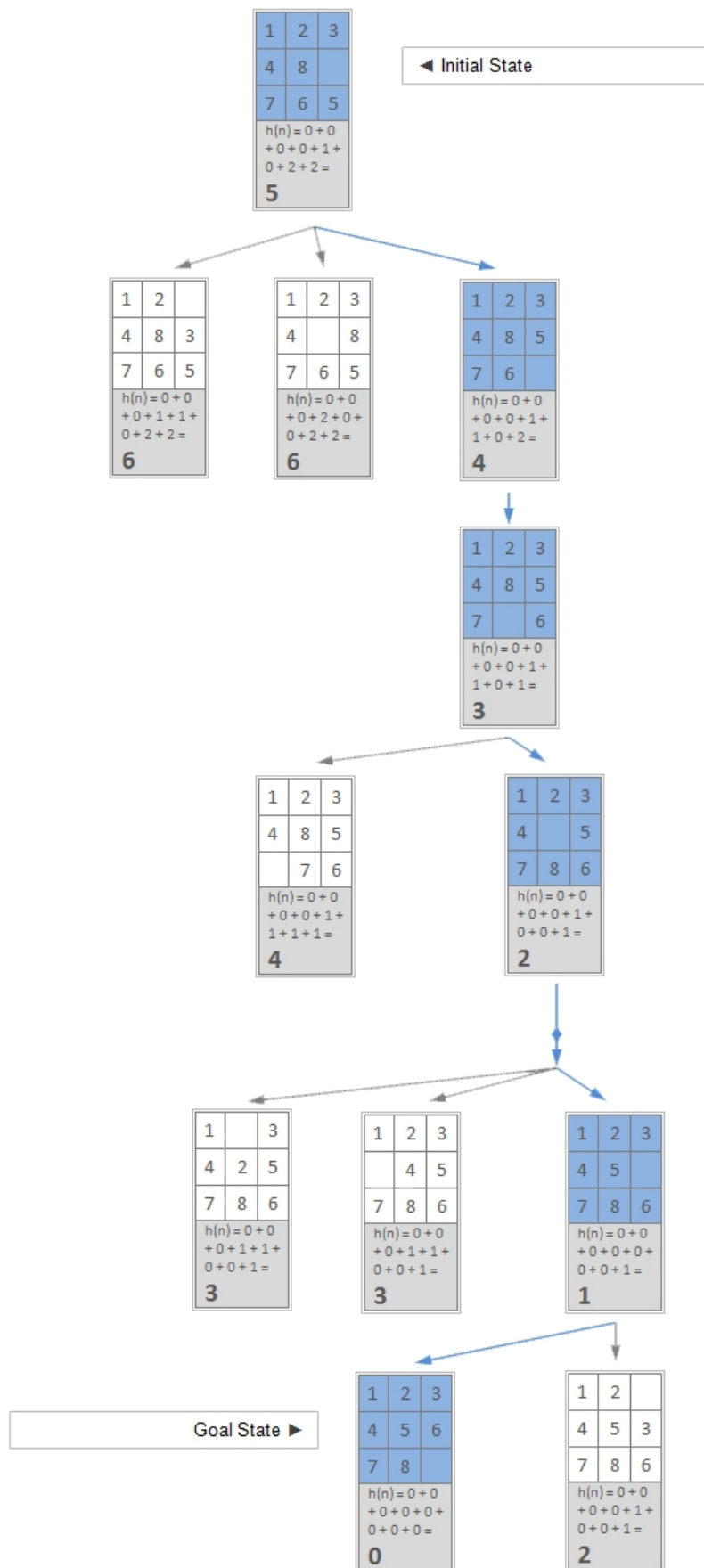
Current State

1	2	3
4	8	
7	6	5

Goal State

1	2	3
4	5	6
7	8	

$h(n)$ adalah nilai total heuristik dari kondisi *puzzle*. Bernilai 0 untuk posisi yang benar dan untuk posisi yang salah nilainya adalah jarak terpendek menuju posisi benar. Proses evaluasi yang dilakukan selalu mengambil nilai heuristik terkecil.



BAB III PENUTUP

3.1 Simpulan

Simpulan dari pembahasan yang telah diuraikan di atas yaitu:

1. *Hill Climbing* adalah proses pengujian yang dilakukan dengan menggunakan fungsi heuristik. *Hill Climbing* terdiri dari dua macam metode yakni *Simple Hill Climbing* dan *Steepest-Ascent Hill Climbing*.
2. Terdapat beberapa langkah dalam algoritma *Simple Hill Climbing* yaitu:
 1. Mulai dari keadaan awal, lakukan pengujian: jika merupakan tujuan, maka berhenti; dan jika tidak, lanjutkan dengan keadaan sekarang sebagai keadaan awal.
 2. Kerjakan langkah-langkah berikut sampai solusinya ditemukan, atau sampai tidak ada operator baru yang akan diaplikasikan pada keadaan sekarang:
 - a. Cari operator yang belum pernah digunakan; gunakan operator ini untuk mendapatkan keadaan yang baru.
 - b. Evaluasi keadaan baru tersebut.
 1. Jika keadaan baru merupakan tujuan, keluar.
 2. Jika bukan tujuan, namun nilainya lebih baik daripada keadaan sekarang, maka jadikan keadaan baru tersebut menjadi keadaan sekarang.
 3. Jika keadaan baru tidak lebih baik daripada keadaan sekarang, maka lanjutkan iterasi.
3. Algoritma *Simple Hill Climbing* pada umumnya diterapkan dalam masalah *Travelling Salesman Problem* (TSP) dan bisa diterapkan untuk menyelesaikan *game Number Puzzle Slider*.

DAFTAR PUSTAKA

Kusumadewi, Sri. Pengantar Kecerdasan Buatan (AK045218):
Teknik Pencarian Heuristik.

Taufiq, Andik. 2010. 8-Puzzle Problem Bagian 2.
<http://andiktaufiq.wordpress.com/2010/05/02/8-puzzle-problem-bagian-2/>.

(Diakses tanggal 25 September 2014)