KECERDASAN BUATAN

Dosen Pengampu : Fitri Nuraeni, S.Kom., M.Kom.

Disusun Oleh:

Dafa Dhaifullah 2206171

Kelas E

PROGRAM STUDI TEKNIK INFORMATIKA

INSTITUT TEKNOLOGI GARUT

2024

1. Algoritma-algoritma yang termasuk dalam blind search antara lain:
   - Depth-First Search (DFS)
   - Breadth-First Search (BFS)
   - Uniform Cost Search (UCS)
   - Iterative Deepening Depth-First Search (IDDFS)

   Sementara itu, algoritma-algoritma yang termasuk dalam heuristic search antara lain:
   - Best-First Search
   - A* Search
   - Greedy Best-First Search
   - Hill Climbing
   - Simulated Annealing
2. Blind search :
   - Depth-First Search (DFS) :

     Mulai dari simpul awal, DFS menelusuri setiap cabang dari simpul secara berurutan hingga mencapai simpul tujuan atau tidak ada cabang lagi yang dapat dijelajahi. DFS menggunakan pendekatan "menyelam dalam" dan menggunakan struktur data tumpukan (stack) untuk menyimpan simpul yang belum dieksplorasi.
   - Breadth-First Search (BFS) :

     Mulai dari simpul awal, BFS menelusuri semua simpul yang bertetangga dengan simpul saat ini sebelum melanjutkan ke simpul-simpul yang lebih jauh. Ini dilakukan dengan menggunakan struktur data antrian (queue) untuk menyimpan simpul yang akan dieksplorasi selanjutnya.
   - Uniform Cost Search (UCS) :

     UCS mencari jalur dengan biaya terendah dari simpul awal ke simpul tujuan. Algoritma ini mempertimbangkan biaya akumulatif setiap jalur dan memilih jalur dengan biaya terendah. UCS juga menggunakan antrian prioritas (priority queue) untuk memilih simpul berikutnya yang akan dieksplorasi.
   - Iterative Deepening Depth-First Search (IDDFS) :

     IDDFS adalah varian dari DFS yang melakukan DFS dengan membatasi kedalaman pencarian pada setiap iterasi. Algoritma ini secara berulang menelusuri simpul-simpul dengan meningkatkan kedalaman maksimal pencarian hingga menemukan solusi atau mencapai batas kedalaman tertentu.

Heuristic search :

- Best-First Search :

  Best-First Search memilih simpul untuk dieksplorasi berdasarkan nilai fungsi evaluasi, yang biasanya diambil dari heuristik. Algoritma ini fokus pada simpul yang memiliki nilai evaluasi terbaik, tanpa memperhatikan jalur yang sudah dilalui.

- A * Search :

  A* Search adalah algoritma pencarian yang kombinasi antara biaya sejauh ini dari simpul awal (g(n)) dan estimasi biaya dari simpul ke tujuan (h(n)). Algoritma ini memilih simpul berdasarkan jumlah biaya g(n) dan h(n), dengan asumsi heuristiknya konsisten.

- Greedy Best-First Search :

  Greedy Best-First Search mirip dengan Best-First Search, namun algoritma ini hanya memperhatikan nilai heuristik (h(n)) untuk memilih simpul berikutnya yang akan dieksplorasi. Tidak memperhitungkan biaya sejauh ini (g(n)).

- Hill Climbing :

  Hill Climbing adalah algoritma optimasi lokal yang berusaha untuk mencapai solusi terbaik dengan memilih langkah yang mengarah ke arah peningkatan terbesar dalam fungsi evaluasi. Namun, bisa terjebak dalam minimum lokal.

- Simulated Annealing :

  Simulated Annealing adalah algoritma pencarian probabilitas yang terinspirasi dari proses pelunakan logam. Algoritma ini menerima solusi buruk dengan probabilitas tertentu untuk menghindari terjebak di minimum lokal, namun kemudian probabilitasnya akan menurun seiring berjalannya waktu.

3. Blind search :
   1) Depth-First Search (DFS) :
      - Time Complexity: O(b^m) (baik kasus terbaik maupun terburuk)
      - Space Complexity: O(bm) (dengan m adalah kedalaman maksimal dan b adalah faktor branching)
      - Completeness: Tidak selalu lengkap karena dapat terjebak dalam loop jika graf memiliki siklus.
      - Optimality: Tidak optimal karena tidak menjamin jalur terpendek.
   2) Breadth-First Search (BFS) :
      - Time Complexity: O(b^d) (dengan d adalah kedalaman solusi dan b adalah faktor branching)

- Space Complexity: O(b^d) (membutuhkan ruang yang besar karena menyimpan semua simpul di level yang sama)
- Completeness: Lengkap jika biaya lintasan adalah seragam.
- Optimality: Optimal jika biaya lintasan seragam, tetapi tidak optimal jika biaya lintasan tidak seragam.

3) Uniform Cost Search (UCS) :
   - Time Complexity: O(b^(C*/ε)) (dengan C* adalah biaya solusi optimal dan ε adalah perbedaan minimum antara biaya dua lintasan)
   - Space Complexity: O(b^(C*/ε))
   - Completeness: Lengkap
   - Optimality: Optimal jika biaya solusi optimal terbatas.
4) Iterative Deepening Depth-First Search (IDDFS) :
   - Time Complexity: O(b^d)
   - Space Complexity: O(bd)
   - Completeness: Lengkap
   - Optimality: Optimal jika biaya seragam dan faktor branching hingga akhir.

Heuristic Search :

1) Best-First Search :
   - Time Complexity: Bergantung pada heuristik yang digunakan.
   - Space Complexity: Bergantung pada heuristik yang digunakan.
   - Completeness: Tergantung pada heuristik dan pemilihan fungsi evaluasi.
   - Optimality: Tidak selalu optimal.
2) A * Search:
   - Time Complexity: O(b^d) dalam kasus terburuk, namun cenderung jauh lebih cepat karena penggunaan fungsi heuristik.
   - Space Complexity: Bergantung pada heuristik, tetapi biasanya membutuhkan lebih sedikit ruang dibanding BFS.
   - Completeness: Lengkap jika fungsi heuristik admissible dan consistent.
   - Optimality: Optimal jika fungsi heuristik admissible dan consistent.
3) Greedy Best-First Search :
   - Time Complexity: Bergantung pada heuristik yang digunakan.
   - Space Complexity: Bergantung pada heuristik yang digunakan.
   - Completeness: Tergantung pada heuristik dan pemilihan fungsi evaluasi.

- Optimality: Tidak optimal.

4) Hill Climbing :
   - Time Complexity: Bergantung pada topologi ruang pencarian dan fungsi evaluasi.
   - Space Complexity: Tergantung pada topologi ruang pencarian.
   - Completeness: Tidak lengkap karena dapat terjebak dalam minimum lokal.
   - Optimality: Tidak optimal karena dapat terjebak dalam minimum lokal.
5) Simulated Annealing :
   - Time Complexity: Bergantung pada parameter simulasi dan strategi penurunan suhu.
   - Space Complexity: Tergantung pada topologi ruang pencarian.
   - Completeness: Tidak lengkap karena bersifat probabilistik.
   - Optimality: Tidak optimal, namun bisa mendekati solusi optimal terutama jika dijalankan dengan parameter yang tepat.