

Tugas 5

Nama : muhammad makhlufi makbullah

Kelas : TK 45 01

NIM : 1103210171

1. Dijkstra's Algorithm

Implementasi & Cara Kerja: Dijkstra's Algorithm mencari jalur terpendek dalam graf berbobot dari titik awal ke titik tujuan. Algoritma ini memastikan bahwa setiap node dikunjungi dengan prioritas berdasarkan jarak kumulatif terkecil dari titik awal, tanpa memanfaatkan heuristic. Semua node dievaluasi hingga titik tujuan ditemukan.

Kelebihan:

- **Optimal dan Lengkap:** Algoritma ini menjamin jalur terpendek akan ditemukan jika jalur tersebut ada.
- **Serbaguna:** Dapat digunakan pada graf dengan bobot yang bervariasi.
- **Akurat pada Kondisi Grid Tanpa Heuristik:** Karena algoritma tidak menggunakan heuristik, solusi yang diberikan akurat dan tidak bergantung pada asumsi jarak.

Keterbatasan:

- **Efisiensi:** Karena mengevaluasi semua node hingga mencapai tujuan, algoritma ini bisa lambat, terutama pada graf besar dan grid dengan banyak node.
- **Komputasi Tinggi pada Grid Besar:** Dalam graf dengan banyak node, kebutuhan komputasinya meningkat drastis.

Situasi Penggunaan Ideal:

- Cocok untuk mencari jalur terpendek dalam graf besar tanpa menggunakan heuristic.
- Ideal digunakan dalam lingkungan tanpa informasi tambahan tentang posisi tujuan relatif terhadap node lainnya.

2. A Algorithm*

Implementasi & Cara Kerja: A* Algorithm menggabungkan pendekatan jalur terpendek dengan fungsi heuristic yang mengarahkan pencarian menuju titik tujuan. Algoritma ini menggunakan fungsi biaya $f(n) = g(n) + h(n)$, di mana $g(n)$ adalah jarak dari titik awal ke node saat ini, dan $h(n)$ adalah estimasi jarak dari node saat ini ke tujuan. Dengan cara ini, A* lebih efisien dalam menemukan jalur, terutama di grid yang memiliki banyak node.

Kelebihan:

- **Efisien dengan Heuristik:** A* sering kali lebih cepat daripada Dijkstra karena memprioritaskan jalur yang lebih dekat dengan tujuan.
- **Optimal:** Jika heuristic yang digunakan sesuai, A* dapat memberikan jalur terpendek.
- **Fleksibel:** Algoritma ini dapat diterapkan dalam berbagai grid, termasuk lingkungan dengan rintangan.

Keterbatasan:

- **Ketergantungan pada Heuristik:** Keefektifan algoritma sangat tergantung pada pilihan heuristic. Heuristic yang tidak sesuai dapat mengurangi efisiensi atau bahkan akurasi.
- **Kompleksitas pada Grid Besar dengan Rintangan:** Ketika menghadapi grid besar atau banyak rintangan, A* masih memerlukan sumber daya komputasi yang signifikan.

Situasi Penggunaan Ideal:

- Cocok untuk mencari jalur terpendek dalam grid besar dengan banyak rintangan, khususnya ketika lokasi tujuan diketahui dengan pasti.
- Ideal untuk lingkungan yang dapat menggunakan heuristic yang sesuai, seperti dalam peta dua dimensi yang membutuhkan jalur langsung ke tujuan.

3. Cell Decomposition

Implementasi & Cara Kerja: Metode Cell Decomposition membagi ruang pencarian ke dalam cell-cell kecil yang merepresentasikan area bebas rintangan. Jalur pencarian kemudian dilakukan di antara cell bebas ini. Trapezoidal atau grid-based decomposition dapat diterapkan untuk grid dua dimensi sederhana, dan metode ini membentuk graf di antara cell bebas tersebut.

Kelebihan:

- **Efisiensi di Lingkungan Kompleks:** Pemecahan area pencarian menjadi cell kecil memungkinkan algoritma untuk fokus pada area bebas rintangan saja, sehingga mengurangi kebutuhan komputasi.
- **Kemampuan Menangani Rintangan Kompleks:** Ideal untuk lingkungan dengan banyak rintangan yang menghalangi jalur langsung.
- **Sederhana dan Mudah Diimplementasikan pada Lingkungan 2D:** Struktur grid membuatnya mudah diterapkan pada ruang yang relatif terbatas.

Keterbatasan:

- **Kebutuhan Pemetaan yang Akurat:** Memerlukan informasi lengkap tentang lingkungan untuk mendefinisikan cell dan rintangan dengan tepat.

- **Efektivitas Menurun pada Grid Besar:** Pada lingkungan yang luas atau dengan resolusi tinggi, cell decomposition memerlukan lebih banyak waktu untuk menyelesaikan pencarian.
- **Kurang Optimal dalam Jalur Detail:** Karena metode ini lebih fokus pada cell besar, detail kecil mungkin terlewatkan, dan jalur di dalam cell tidak selalu optimal.

Situasi Penggunaan Ideal:

- Lingkungan dengan rintangan kompleks dan area pencarian yang padat.
- Cocok untuk perencanaan dalam ruang yang terbatas, di mana cell decomposition dapat memberikan jalur yang aman di antara rintangan.

Kesimpulan Umum

Ketiga algoritma ini memiliki kelebihan dan kelemahan masing-masing, yang membuatnya lebih cocok untuk berbagai situasi:

- **Dijkstra's Algorithm** sangat cocok untuk menemukan jalur terpendek tanpa bergantung pada heuristic, dan digunakan saat graf kecil atau ketika jalur yang pasti optimal sangat penting.
- *A* Algorithm** efektif pada grid besar dengan rintangan dan heuristic yang sesuai, mempercepat pencarian jalur melalui estimasi jarak ke tujuan.
- **Cell Decomposition** lebih cocok dalam lingkungan dua dimensi dengan rintangan kompleks dan memungkinkan pencarian jalur di antara cell-cell bebas hambatan.

Dengan mempertimbangkan kekuatan dan kelemahan masing-masing algoritma, pemilihan algoritma yang sesuai dapat membantu meningkatkan efisiensi dan akurasi dalam pencarian jalur pada berbagai lingkungan dan kebutuhan perencanaan robotik.