

ANALISA PEMECAHAN MASALAH “CROSS THE BRIDGE PUZZLE”

Muhammad Fikri Shahabudin(13505016)

Program Studi Teknik Informatika Institut Teknologi Bandung
Jl. Ganesha 10, Bandung.
e-mail: if15016@students.if.itb.ac.id

ABSTRAK

Backtracking, Branch and Bound, Greedy dan Brute Force adalah teknik-teknik yang digunakan dalam menemukan solusi dalam permasalahan untuk banyak bidang. Teknik-teknik ini dapat diaplikasikan dalam games atau permainan, misalnya puzzle. Dalam hal ini akan dibandingkan kemangkusan masing-masing algoritma dalam menyelesaikan puzzle “Cross the Bridge Puzzle”. Tujuan utama dari puzzle ini adalah memindahkan sebuah keluarga yang terdiri dari lima orang dari satu tebing awal ke tebing tujuan melalui sebuah jembatan dengan batasan-batasan tertentu, setiap anggota keluarga yang akan menyeberang juga memiliki karakteristik yang berbeda satu sama lainnya, dilihat dari banyaknya waktu yang dibutuhkan untuk menyeberang jembatan. Dalam hal ini akan dianalisis algoritma algoritma yang memberikan solusi, Backtracking, Branch and bound, greedy, dan brute force. Backtracking adalah algoritma yang berdasarkan kepada algoritma DFS (Deep First Search), menggunakan fungsi pembatas dan pembangkit. Sebaliknya Branch and Bound mengambil pendekatan BFS (Breadth First Search), brute force adalah algoritma yang kurang pintar, namun kelebihanannya adalah hampir semua persoalan dapat diselesaikan dengan cara ini, sementara Greedy adalah Algoritma yang membentuk solusi langkah per langkah (*step by step*). Oleh karena itu, pada setiap langkah harus dibuat keputusan yang terbaik dalam menentukan pilihan.

Kata kunci: Backtracking, Branch and Bound, Plastelina Games, Brute Force.

1. PENDAHULUAN

Puzzle “Cross the Bridge” ini dapat diunduh dari url <http://www.plastelina.net/games/game3.html>. Berikut ini adalah deskripsi yang diberikan pada situs tersebut : “Tolonglah sebuah keluarga untuk mencapai sisi lain dari

jembatan. Perhatikanlah bahwa saat ini adalah malam hari, dan hanya terdapat sebuah lampu. Maksimum 2 (dua) orang dapat menyeberang pada waktu bersamaan dan lampu tersebut harus ada pada saat menyeberang, dengan kata lain seseorang harus menjemput anggota keluarga yang lain agar seluruh anggota keluarga dapat menyeberang, setiap anggota keluarga menyeberang dengan kecepatan yang berbeda-beda, masing-masing 1,3,6,8, dan 12 hitungan waktu. Perlu diperhatikan bahwa jika dua anggota keluarga menyeberang secara bersamaan maka anggota yang lebih cepat akan berjalan dengan kecepatan anggota yang lebih lambat. Lampu akan padam dalam 30 hitungan waktu.



Gambar 1. “Cross The Bridge Games”.

2. METODE

Terdapat beberapa metode yang digunakan dalam pemecahan puzzle diatas, diantaranya dengan Algoritma Brute Force, Algoritma Runut Balik atau Backtracking, Algoritma Branch and Bound, dan Brute Force dan Greedy.

2.1 Penyelesaian dengan Algoritma Brute Force

Sesuai dengan namanya, algoritma brute force adalah algoritma yang kurang pintar, namun kelebihanannya adalah hampir semua persoalan dapat diselesaikan dengan cara ini. Dengan cara ini kita mencari solusi dengan mencoba seluruh kemungkinan yang ada. Pada kasus menyeberang jembatan ini seluruh kemungkinan yang harus diperiksa sebanyak

$$\begin{aligned} S &= C(5,2) \times C(2,1) \times C(4,2) \times C(3,1) \times C(3,2) \times C(4,1) \quad (1) \\ &= 10 \times 2 \times 6 \times 3 \times 3 \times 4 \\ &= 4320 \end{aligned}$$

Pertama – tama dari lima orang anggota keluarga, dua orang menyeberang dari tebing awal ke tebing tujuan, sebutlah dua orang ini pasangan X, dengan membawa lampu, sehingga kemungkinan seluruh pasangan yang ada adalah $C(5,2)$. Selanjutnya seseorang akan membawa lampu kembali ke seberang awal, karena jumlah orang yang ada di seberang tujuan saat itu hanyalah pasangan X, maka kemungkinannya kita kalikan $C(2,1)$. Selanjutnya kemungkinan pasangan kedua, Pasangan Y dapat dipilih dari empat orang sehingga kita kalikan dengan $C(4,2)$. Kini di seberang tujuan terdapat tiga orang, dan salah seorang harus membawa kembali lampu ke tebing asal maka kita kalikan dengan $C(3,1)$. Di tebing asal kita ambil sepasang dari tiga orang yang ada disana, kalikan dengan $C(3,2)$. Terakhir karena ada empat orang di tebing tujuan kita kalikan dengan $C(4,1)$. kini di tebing asal hanya terdapat satu pasangan terakhir yang akan menyeberang. Terdapat enam solusi untuk permasalahan ini, akan dijelaskan kemudian.

2.2 Penyelesaian dengan prinsip Greedy

Greedy adalah Algoritma yang membentuk solusi langkah per langkah (*step by step*). Oleh karena itu, pada setiap langkah harus dibuat keputusan yang terbaik dalam menentukan pilihan. Keputusan pada langkah sebelumnya tidak dapat diganti pada langkah selanjutnya. Pendekatan yang digunakan di dalam Algoritma greedy adalah membuat pilihan yang “tampaknya” memberikan perolehan terbaik.

Algoritma greedy menerapkan dua prinsip sebagai berikut :

1. Mengambil pilihan terbaik pada saat ini tanpa mempertimbangkan langkah selanjutnya (“*take what you can get now*”). Disebut optimum lokal.
2. Berharap dengan optimum lokal akan mengarah ke optimum global.

Pada kasus “*cross the bridge*” ini, prinsip greedynya adalah prioritas sebagai berikut :

1. Memilih anggota terkecil untuk mengembalikan lampu dari tebing tujuan ke tebing asal.
2. Mengusahakan anggota yang berpasangan memiliki selisih waktu yang tidak terlalu besar.

Langkah – langkah dengan Algoritma Greedy untuk kasus diatas adalah :

Langkah 1 : Berdasarkan prinsip 1, maka kita harus mengusahakan agar anggota dengan waktu 1 satuan waktu ada di tebing tujuan. Dan berdasarkan prinsip 2, maka pasangan yang pertama menyeberang adalah (1,3). Waktu tersisa = $30 - \max(3,1) = 27$.

Langkah 2 : Berdasarkan prinsip 1, maka kita pilih anggota dengan waktu 1 satuan waktu. Waktu Tersisa = $27 - 1 = 26$.

Langkah 3 : selanjutnya, kita pilih (2,8). Waktu Tersisa = $26 - 12 = 14$.

Langkah 4 : selanjutnya, kita pilih 3. Waktu Tersisa = $14 - 3 = 11$.

Langkah 5 : selanjutnya, kita pilih (1,3). Waktu Tersisa = $11 - 3 = 8$.

Langkah 6 : selanjutnya, kita pilih 1. Waktu Tersisa = $8 - 1 = 7$.

Langkah 7 : selanjutnya, kita pilih (1,6). Waktu Tersisa = $7 - 1 = 6$.

Solusi didapat dengan tujuh langkah, dengan sisa waktu tersisa satu satuan waktu.

2.3 Penyelesaian dengan Algoritma Backtracking

Runut-balik, yang merupakan perbaikan dari algoritma brute-force, secara sistematis mencari solusi persoalan di antara semua kemungkinan solusi yang ada. Dengan metode runut-balik, kita tidak perlu memeriksa semua kemungkinan solusi yang ada. Hanya pencarian yang mengarah ke solusi saja yang selalu dipertimbangkan.

Solusi persoalan dinyatakan dalam vector 7 (tujuh) tuple:

$$S = [X_1, X_2, X_3, X_4, X_5, X_6, X_7] \quad (1)$$

$$S = [(x_1, x_2), x_3, (x_4, x_5), x_6, (x_7, x_8), x_9, (x_{10}, x_{11})] \quad (2)$$

$x_i = 1, 3, 6, 8$ dan 12 (berdasarkan waktu penyebrangan)

elemen X dengan urutan ganjil terdiri dari dua elemen menggambarkan dua anggota yang menyebrang secara bersamaan, sementara elemen X dengan urutan genap hanya terdiri dari satu elemen menggambarkan anggota keluarga yang mengembalikan lampu untuk kemudian digunakan oleh anggota keluarga lain.

Mis: $[(1,3), 1, (8,12), 3, (1,6), 1, (1,3)]$

Penjelasannya adalah anggota keluarga dengan kecepatan 1 dan 3 (selanjutnya hanya akan disebut anggota 1 atau 3 saja) menyebrang secara bersamaan, kemudian anggota 1 kembali dengan membawa lampu, selanjutnya anggota 8 dan 12 menyebrang, lalu anggota 3 yang telah berada disebatang membawa lampu kembali ke seberang yang lain, begitu seterusnya. Karena bila berjalan secara berpasangan anggota yang lebih cepat mengikuti waktu dari anggota yang lebih lambat maka waktu yang dibutuhkan adalah $3 + 1 + 12 + 3 + 6 + 1 + 3 = 29$ satuan waktu. Dalam hal ini contoh tersebut merupakan salah satu solusi karena memenuhi fungsi batas yang ditetapkan sebagai berikut :

$$B(X_1, X_2, \dots, X_k), X_1 + X_2 + \dots + X_k \leq 30 \quad (3)$$

Untuk X_i yang terdiri dari dua elemen (x_1, x_2) maka nilai X_i sama dengan nilai anggota terbesarnya. Batasan lainnya adalah jika waktu yang tersisa kurang dari penjumlahan waktu dari anggota-anggota yang berada pada tebing awal. Karena yang menyeberang adalah pasangan dan waktu pasangan adalah waktu anggota yang lebih besar maka waktu tersisa harus lebih kecil dari penjumlahan anggota terbesar yang berada pada tebing awal dengan urutan pertama, ketiga, dan kelima. K adalah anggota yang berada pada tebing awal. K_3 dan K_5 mungkin saja bernilai nol jika anggota yang ada pada tebing tujuan kurang dari 3 atau 5.

$$W_t \leq K_1 + K_3 + K_5 \quad (4)$$

Berbeda dengan algoritma brute force, Runut Balik lebih mangkus karena melakukan pruning pada ruang pohon status sehingga dari 4320 kemungkinan pada metoda bruteforce dapat diperkecil.

Terdapat enam buah solusi yang didapat dari kasus "cross the bridge" dinyatakan dalam persamaan (2) yaitu :

Solusi 1 : $[(1,3), 1, (8,12), 3, (1,3), 1, (1,6)]$

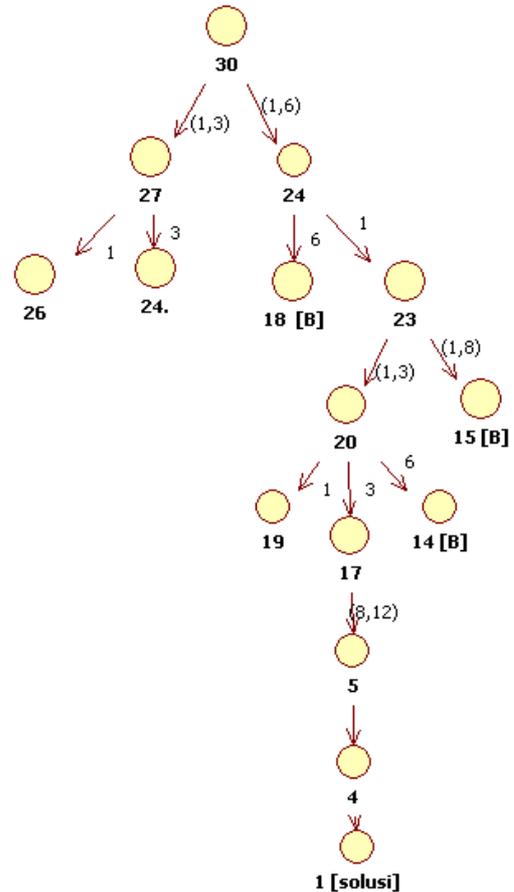
Solusi 2 : $[(1,3), 1, (8,12), 3, (1,6), 1, (1,3)]$

Solusi 3 : $[(1,3), 3, (8,12), 1, (1,3), 1, (1,6)]$

Solusi 4 : $[(1,3), 3, (8,12), 1, (1,6), 1, (1,3)]$

Solusi 5 : $[(1,6), 1, (1,3), 1, (8,12), 3, (1,3)]$

Solusi 6 : $[(1,6), 1, (1,3), 3, (8,12), 1, (1,3)]$



Gambar 2. Sebagian dari Pohon yang terbentuk selama Backtracking.

Salah satu dari enam buah solusi yang didapat. Hanya diperlihatkan satu solusi karena pohon yang cukup besar sehingga sulit untuk merepresentasikan secara menyeluruh.

2.4 Penyelesaian dengan Algoritma Branch and Bound

Sebagaimana algoritma runut – balik, algoritma *Branch and Bound* (B&B) juga merupakan metoda pencarian di dalam ruang solusi secara sistematis. Ruang solusi diorganisasikan kedalam pohon ruang status Pembentukan pohon ruang status pada algoritma branch and bound berbeda dengan algoritma Runut – Balik. Algoritma

Runut – Balik menerapkan prinsip DFS (Depth First Search) sementara B&B menerapkan prinsip BFS.

Untuk mempercepat pencarian ke simpul solusi , setiap simpul diberi ongkos (cost). Pada penerapannya B&B tidak lagi merupakan BFS murni karena pembangkitan simpul berikutnya adalah yang costnya paling rendah, tidak lagi berdasarkan urutan.

Pada dasarnya dalam kasus ini, batasannya tidak terlalu berbeda dengan pada algoritma runut balik, namun berdasarkan pencarian dengan algoritma runut balik solusi –solusi berada pada simpul – simpul awal yang dalam sehingga dapat disimpulkan dalam pencarian ini DFS lebih mangkus ketimbang BFS. Sehingga Runut Balik lebih baik diterapkan dalam kasus ini.

IV. KESIMPULAN

Pada makalah ini penulis mencoba memberikan penyelesaian terhadap permasalahan “cross the bridge” puzzle.sebagian dari sekian banyak algoritma yang memberikan solusi antara lain, Backtracking, Branch and bound, greedy, dan brute force . Backtracking adalah algoritma yang berdasarkan kepada algoritma DFS (Deep First Search), menggunakan fungsi pembatas dan pembangkit. Sebaliknya Branch and Bound mengambil pendekatan BFS (Breadth First Search), brute force adalah algoritma yang kurang pintar, namun kelebihanannya adalah hampir semua persoalan dapat diselesaikan dengan cara ini,sementara Greedy adalah Algoritma yang membentuk solusi langkah per langkah (*step by step*). Oleh karena itu, pada setiap langkah harus dibuat keputusan yang terbaik dalam menentukan pilihan. Runut Balik lebih baik daripada B&B dan Brute Force, sementara pada algoritma greedy agak sulit menentukan fungsi layak untuk optimum lokal, sehingga fungsi untuk optimum lokal cenderung tidak konsisten atau berubah – ubah agar dapat dicapai solusi.

REFERENSI

- [1] <http://www.plastelina.net/games/game3.html>
Diakses tanggal 18 Mei 2007 pukul 13.00
- [2] Munir, Rinaldi “Strategi Algoritmik”, Institut Teknologi Bandung, 2007.