

Perbandingan Penggunaan Algoritma Genetik dan Teori Graf pada Simulasi Game Fighting

Fitrandi Ramadhan 13508065
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
If18065@students.if.itb.ac.id

ABSTRAK

Fighting game merupakan game yang memiliki keberagaman yang luas. Mengapa saya katakan bahwa genre game ini memiliki keberagaman yang luas? Karena pada game ini anda tidak akan menemukan 2 buah pertarungan yang sama dimana aksi yang terjadi tidak akan sama dengan pertarungan sebelumnya. Ini karena dalam genre game fighting biasanya player memiliki keluwesan gerakan yang tinggi. Tidak hanya itu, sedikit saja perbedaan gerakan yang dilakukan oleh seorang player, ini akan mengakibatkan efek yang terjadi pada sang lawan akan berbeda pula. Pada game fighting ini player akan melancarkan sejumlah serangan kepada lawannya, tidak hanya satu serangan saja tentunya. Rangkaian serangan inilah yang akan kita bahas dan selanjutnya akan kita sebut sebagai Combo.

1. PENDAHULUAN

Game fighting memiliki keluwesan gerakan yang tinggi dalam genre game fighting ini maka combo-combo yang bisa dilancarkan pun akan bervariasi sesuai dengan keadaan player dan musuh. Dalam eksekusi combo ini biasanya musuh akan terpental atau terdisable (player musuh tidak dapat mengendalikan character nya karena character nya terpental ke udara dst). Dikarenakan keadaan yang sangat bervariasi inilah sulit untuk mengeksekusi combo-combo tersebut secara sukses. Karena itu sulit pula menentukan combo apakah yang paling panjang. Panjang dalam artian disini adalah combo yang memiliki waktu disable yang paling lama, dan menghasilkan damage output yang paling besar.

Karena rangkaian combo yang sangat bervariasi ini dan keadaan environment di dalam game yang sangat tidak stabil membuat kesulitan player untuk menciptakan comb sendiri yang paling efisien tersebut. Oleh karena itu disini saya ingin menciptakan sebuah combo yang paling efisien dengan environment yang dapat dikontrol. Yaitu dengan mensimulasikanya dengan algoritma genetik. Game yang

saya gunakan sendiri adalah game Street Fighter vs Tekken.

Algoritma Genetik adalah sebuah algoritma searching dimana prosesnya mengikuti proses evolusi makhluk hidup. Algoritma ini biasanya digunakan untuk Optimasi yang mana sangat cocok dengan simulasi game fighting ini karena dibutuhkan optimasi dala menemukan combo yang efisien seperti yang saya sebutkan diatas. Algoritma genetic ini menggunakan teori- teori seperti inheritance, mutation, selection dan crossover.

Dalam simulasi ini dimodelkanlah satu buah serangan sebagai sebuah gen dan untuk kemudian dirangkai emdnjadi combo yang dilakukan akan dimodelkan sebagai kromosom nya dimana satu rantai kromosom merupakan rangkaian gerakan (serangan) yang akan digenerate secara terus menerus sesuai dengan Algoritma. Kemudian akan dibuat sebuah fitness function dimana nantinya kromosom ini akan dievaluasi. Evaluasi fitness function ini yaitu dengan melihat disable time dan damage output yang dihasilkan dari rantai kromosom tersebut. Kemudian rangkaian kromosom ini akan di silangkan dengan kromosom lain untuk nantinya membentuk sebuah kromosom baru yang siap dievaluasi lebih lanjut.

Setelah selesai sejumlah iterasi, akan dapat dihasilkan bahwa suatu combo merupakan suatu combo yang ideal dimana combo tersebut memiliki disable time dan damage output yang paling besar dibanding dengan combo lainnya. Dan combo inilah yang diinginkan sebagai sebuah output untuk pada nantinya dapat diaplikasikan di Environment Game.

Karena penggunaan algoritma genetic diatas sangatlah rumit dan dibutuhkan simulasi yang intensif saya ingin menyederhanakan nya dengan sebuah graf pohon sebagai decision tree.

2. METODE

Metode yang digunakan penulis adalah studi literature sebagai sumber utama dan percobaan langsung sebagai

pendukung referensi penulis dalam membuat makalah ini. Tidak lupa penulis selalu mencatat berbagai kemungkinan yang terjadi pada permainan ini.

3. Teori Dasar

Ada empat langkah yang harus dilakukan untuk mendapatkan hasil dengan menggunakan genetic algorithm yaitu :

- Inisialisasi : Proses pembuatan new population
- Seleksi : Proses pemilihan populasi untuk mendapatkan populasi yang lebih baik
- Reproduksi : Proses untuk mengenerate populasi anak yang lebih bervariasi
- Terminasi : Pemberhentian dari proses daur seleksi dan reproduksi sampai suatu titik tertentu

Inisialisasi pada simulasi ini dilakukan dengan mengenerate suatu rantai kromosom secara random. Gen yang digunakan pada rantai kromosom ini adalah suatu pola serangan yang dapat dilakukan oleh player dengan penjelasan contoh pola serangan seperti pada table dibawah ini.

Attack	ID	Damage	Count	Next Possible Move
Low Punch	PL	1	1	PM, PH, KL, KM, KH
Medium Punch	PM	2	1	PH, KH
High Punch	PH	3	2	PH
Low Kick	KL	1	1	PM, PH, KM, KH
Medium Kick	KM	2	1	PH, KH
High Kick	KH	3	2	KH



Low Punch



High Kick

Kemudian dibuatlah rantai serangan sesuai dengan aturan diatas. Next Possible Move ini adalah hasil dari pengambilan data manual dari masing-masing character game dimana setiap character memiliki jenis attack, count, damage, dan tentunya next possible move yang berbeda-beda. Tabel diatas dibuat berdasarkan character JIN, yang mana nanti akan digunakan sebagai specimen simulasi ini. Contoh dari rantai serangan adalah sebagai berikut.

PL	KL	PM	PH	PH	KH	KH
----	----	----	----	----	----	----

Rantai serangan yang dihasilkan pada proses inisialisasi inilah yang akan menjadi populasi awal untuk diseleksi lebih lanjut.

Seleksi dilakukan untuk mendapatkan suatu rantai kromosom yang paling optimal. Dalam kasus ini yaitu suatu rantai kromosom yang menghasilkan damage output paling besar. Adapun rumusan fitness functionnya sebagai berikut.

$$f(x) = A_1 + A_2 + A_3 + \dots + A_n$$

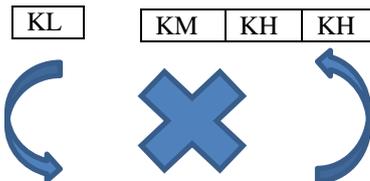
Dua rantai kromosom akan dibandingkan hasilnya. Dimana suatu rantai yang menghasilkan value yang lebih kecil akan dihapus(Punah) dan rantai kromosom yang memberikan hasil lebih besar akan melakukan reproduksi.

Apabila pada dua buah rantai kromosom menghasilkan hasil tertinggi pada suatu evaluasi. Maka dua kromosom tersebut dianggap bersaing dan dilakukan crossover keduanya pada proses reproduksi.. Proses pada simulasi ini dilakukan dengan aturan 2 buah rantai serangan akan dicari potongan yang sesuai untuk digabungkan dengan mematuhi Next Possible Move kemudian potongan tersebut saling bertukar.

KL	PM	PH	PH
----	----	----	----



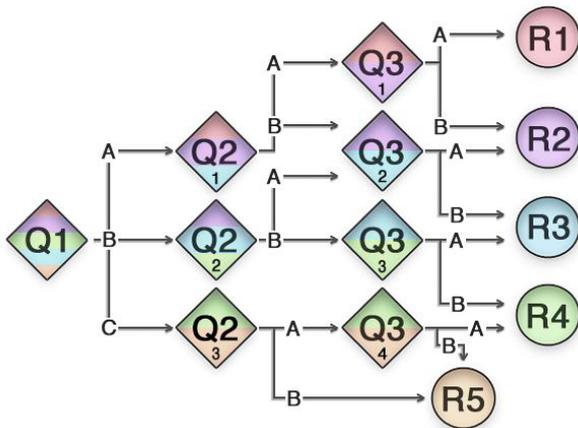
KL	KM	KH	KH
----	----	----	----



KL	KM	PM	PH	PH
----	----	----	----	----

Untuk memberhentikan daur proses genetic algorithm ini. State terminasi yang dilakukan adalah dengan membatasi secara manual berapa kali proses iterasi dari genetic algorithm ini akan dilakukan. Dan individu yang survive di akhir iterasi dinyatakan sebagai hasil optimum dari genetic algorithm ini.

Dengan penggunaan decision tree akan dapat dimodelkan menjadi sebuah pohon yang berisikan berbagai kemungkinan serangan yang dilakukan.



Referensi gambar

Setiap keputusan tetap harus diperiksa kembali dengan menggunakan simulasi.

4. Simulasi dan Analisis

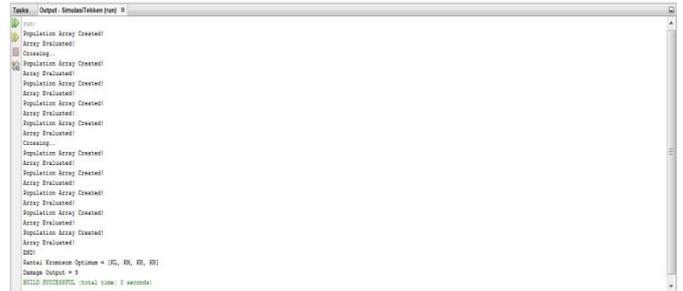
Simulasi ini dilakukan pada game Street Fighter X Tekken dengan character yang digunakan adalah Jin Kazama. Pemilihan Jin Kazama sebagai specimen karena character ini memiliki pola serangan yang paling standar dengan damage yang dihasilkan/serangan medium dan rata-rata kombo moderat.

Pada Inisialisasi hasil pengambilan data yang dilakukan pada Jin Kazama.

Attack	ID	Damage	Count	Next Possible Move
Low Punch	PL	1	1	PM, PH, KL, KM, KH
Medium Punch	PM	2	1	PH, KH
High Punch	PH	3	2	PH
Low Kick	KL	1	1	PM, PH, KM, KH
Medium Kick	KM	2	1	PH, KH
High Kick	KH	3	2	KH

Tabel Jin Kazama

Hasil Simulasi yang didapat sebagai berikut.



Simulasi 1



Simulasi 2

Dapat dilihat di dua hasil simulasi diatas bahwa output yang dihasilkan berbeda dimana pada simulasi pertama didapatkan hasil Rantai Kromosom Optimum = [KL, KM, KH, KH] dengan Damage Output 9. Sedangkan pada simulasi kedua Rantai Kromosom Optimum = [PL, KL, PM, KH, KH] dengan Damage Output 10.

Perbedaan hasil yang didapat ini terjadi karena randomasi yang dilakukan oleh program sangatlah acak. Dimana tidak dimasukkannya rantai yang sudah pernah digenerate kedalam suatu memori sehingga terkadang membuat redundansi Array Move yang ada. Dikarenakan proses cross over hanya akan terjadi apabila ada rantai Kromosom yang memiliki hasil fitness function sama maka proses crossover ini tidak terjadi di setiap iterasi algoritma genetic. Pada simulasi ini hanya dilakukan proses iterasi algoritma genetic dengan 10 kali iterasi. Hal ini dilakukan karena memang kombinasi serangan yang dapat dilakukan pada game ini terhitung sangat sedikit dibandingkan dengan game fighting lainnya.

Kekurangan simlulasi ini yaitu pada redundansi Array Move yang degenerate dapat ditutup. Seperti pada pengaplikasian algoritma genetic lainnya. Semakin banyak iterasi yang dilakukan maka akan semakin baik pul hasil Optimum yang dihasilkan. Walaupun tanpa jaminan bahwa hasil yang dikeluarkan adalah hasil yang paling optimum namun dengan peningkatan iterasi sudah pasti aka nada peningkatan kualitas dari output.

Output yang dihasilkan pada simulasi ini dicoba kembali di dalam game dan hasil yang didapat sangatlah memuaskan karena memang inilah Kombo serangan yang paling optimum dilakukan pada game ini.



1st Move



2nd Move



3rd Move



4th Move



5th Move



6th Move

Decision Tree dapat diselesaikan dengan algoritma DFS atau BFS . Kedua algoritma ini cocok dilakukan apabila rantai combo memiliki panjang yang kecil. Apabila panjang combo sudah cukup besar maka penyelesaiannya akan memiliki waktu operasi yang sangat lama.

5. Konklusi

Konklusi yang didapat adalah dimana sebuah kombinasi serangan dalam game fighting dapat dimodelkan menjadi sebuah Array untuk kemudian diproses dengan algoritma genetic yang mana menghasilkan sebuah rangkaian serangan optimal. Simulasi yang dilakukan kali ini hanyalah pada game Street Fighter Vs Tekken akan tetapi simulasi dapat dilakukan pada game fighting apapun karena pada game-game fighting hanyalah bentuk-bentuk serangan yang dapat dilancarkan. Bahkan sampai pada game yang memiliki variasi sampai ratusan sekalipun. Hal yang harus disesuaikan dengan simulasi adalah Pengambilan data yang perlu diambil dari masing-masing game tersebut baik secara manual(yang dilakukan pada simulasi ini) ataupun otomatis. Kebanyakan game fighting memiliki variasi input yang beragam mulai dari kombinasi tombol arah sampai dengan penekanan 2 sampai 4 tombol sekaligus. Maka penanggulangan kekurangan untuk mendapatkan Rantai serangan yang Optimal adalah yang ada pada simulasi ini adalah dengan iterasi yang lebih besar. Atau dapat juga dibuat sebuah parameter batas minimum damage output untuk terminasi simulasi tersebut.

Penggunaan decision Tree menjadi sangat lah efisien bila panjang rantai relatif kecil sehingga waktu operasi bisa dilakukan dengan cepat. BFS akan menghasilkan hasil yang lebih efisien dengan waktu yang lebih lama.

REFERENSI

- [1] Jonathan Byrne, Michael O'Neill, Anthony Brabazon. Optimizing Offensive Moves in Toribash Using Genetic Algorithm. University College Dublin.
- [2] TP Hong, KY Huang, WY Lin. Applying Genetic Algorithm to Game Search Tree. 2002
- [3] Ian Watson, Damir Azhar, Ya Chuyang, Wei Pan & Gary Chen. Optimization in Strategy Games: Using Genetic Algorithms to Optimize City Development in FreeCiv.
- [4] William H. Hsu. Genetic Algorithm. Department of Computing and Information Sciences. Kansas State University.
- [5] Dr. Hamid Nemati. Genetic Algorithms
- [6] <http://c2.com/cgi/wiki?GeneticAlgorithm>. accessed July 2012
- [7] http://manuals.matrix.squizzesuite.net/_images/other-cms-assets/decision-tree/decision-tree-diagram.png