

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

Who? Michael Adrian
2013730039
michaeladrian39@gmail.com

From? Jurusan Teknik Informatika
Fakultas Teknologi Informasi dan Sains
Universitas Katolik Parahyangan

When? 6 Desember 2016

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

Who? Michael Adrian
2013730039
michaeladrian39@gmail.com

From? Jurusan Teknik Informatika
Fakultas Teknologi Informasi dan Sains
Universitas Katolik Parahyangan

When? 6 Desember 2016

Calcudoku

- Salah satu jenis permainan teka-teki aritmatika dan logika
- Dikenal juga sebagai KenKen, KenDoku, atau Mathdoku
- Diciptakan pada tahun 2004 oleh Tetsuya Miyamoto, seorang guru matematika dari Jepang
- Diciptakan untuk melatih kemampuan matematika dan logika dengan cara yang menyenangkan

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

- Dasar Teori

—Calcudoku

—Calcudoku

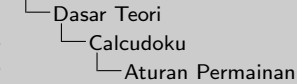
- Salah satu jenis permainan teka-teki aritmatika dan logika
- Dikenal juga sebagai KenKen, KenDoku, atau Mathdoku
- Diciptakan pada tahun 2004 oleh Tetsuya Miyamoto, seorang guru matematika dari Jepang
- Diciptakan untuk melatih kemampuan matematika dan logika dengan cara yang menyenangkan

Sebagai salah satu jenis permainan teka-teki aritmatika dan *grid*, Calcudoku, atau dikenal juga sebagai KenKen, KenDoku, atau Mathdoku, diciptakan pada tahun 2004 oleh seorang guru matematika dari Jepang yang bernama Tetsuya Miyamoto untuk memenuhi tujuannya untuk melatih kemampuan matematika dan logika siswa-siswinya dengan cara yang menyenangkan. Nama KenKen diambil dari kata bahasa Jepang yang berarti kepandaian. Permainan yang mengasah otak ini dengan cepat menyebar ke seluruh Jepang dan Amerika Serikat, menggantikan permainan teka-teki silang di banyak koran. Permainan ini kemudian menjadi sensasi di seluruh dunia setelah munculnya versi *online* dan *mobile* dari permainan teka-teki ini, khususnya menarik untuk pecinta permainan teka-teki angka seperti Sudoku.

Aturan Permainan

- Pemain diberikan sebuah *grid* dengan ukuran $n \times n$
- n biasanya antara 3 sampai dengan 9
- *Grid* ini harus diisi dengan angka 1 sampai dengan n
- Dalam setiap baris setiap angka hanya muncul sekali
- Dalam setiap kolom setiap angka hanya muncul sekali
- *Grid* dibagi ke dalam *cage*
- *Cage* adalah sekelompok sel yang dibatasi oleh garis yang lebih tebal daripada garis pembatas antar sel dengan angka tujuan dan operator yang telah ditentukan
- Angka-angka dalam setiap *cage* harus mencapai angka tujuan jika dihitung menggunakan operator yang telah ditentukan
- Angka tujuan dan operasi yang telah ditentukan ditulis di sudut kiri atas *cage*

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku



Seperti dalam Sudoku, dalam teka-teki ini, pemain diberikan sebuah *grid* dengan ukuran $n \times n$, dengan n biasanya antara 3 sampai dengan 9. *Grid* ini harus diisi dengan angka 1 sampai dengan n sehingga dalam setiap baris setiap angka hanya muncul sekali, dalam setiap kolom setiap angka hanya muncul sekali. Perbedaannya dengan Sudoku adalah, Calcudoku dibagi ke dalam *cage* (sekelompok sel yang dibatasi oleh garis yang lebih tebal daripada garis pembatas antar sel dengan angka tujuan dan operator yang telah ditentukan), dan angka-angka dalam setiap *cage* harus mencapai angka tujuan jika dihitung menggunakan operator yang telah ditentukan. Angka tujuan dan operasi yang telah ditentukan ditulis di sudut kiri atas *cage*.

- Pemain diberikan sebuah grid dengan ukuran $n \times n$
- n bilangan antara 3 sampai dengan 9
- Isi grid ini harus diisi dengan angka 1 sampai dengan n
- Dalam setiap baris setiap angka hanya muncul sekali
- Dalam setiap kolom setiap angka hanya muncul sekali
- Grid dibagi ke dalam cage
- Cage adalah sekelompok sel yang dibatasi oleh garis yang lebih tebal daripada garis pembatas antar sel dengan angka tujuan dan operator yang telah ditentukan
- Angka-angka dalam setiap cage harus mencapai angka tujuan jika dihitung menggunakan operator yang telah ditentukan
- Angka tujuan dan operasi yang telah ditentukan ditulis di sudut kiri atas cage

Operator-Operator Matematika

- Ada 5 kemungkinan operator:
 - $+$ (penjumlahan)
 - $-$ (pengurangan)
 - \times (perkalian)
 - \div (pembagian)
 - $=$ (sama dengan)
- Jika operasi matematika yang ditentukan adalah pengurangan atau pembagian, maka ukuran *cage* harus berukuran dua sel

Ada lima kemungkinan operator:

1. $+$, sebuah operator n -ary yang menandakan penjumlahan.
2. $-$, sebuah operator biner yang menandakan pengurangan.
3. \times , sebuah operator n -ary yang menandakan perkalian.
4. \div sebuah operator biner yang menandakan pembagian.
5. $=$, (simbol ini biasanya dihilangkan), sebuah operator uner yang menandakan persamaan.

Jika operasi matematika yang ditentukan adalah pengurangan atau pembagian, maka ukuran *cage* harus berukuran dua sel. Pada beberapa versi dari teka-teki ini, hanya angka tujuan yang diberikan, dan pemain harus menebak operator dari setiap *cage* untuk menyelesaikan teka-tekinya.

- Ada 5 kemungkinan operator:
 - + (penjumlahan)
 - - (pengurangan)
 - × (perkalian)
 - ÷ (pembagian)
 - = (sama dengan)
- Jika operasi matematika yang ditentukan adalah pengurangan atau pembagian, maka ukuran cage harus berurutan dua sel

Contoh Permainan

3	8 +	3 -	
7 +			
	8 +	8 +	
			1

Figure 1: Contoh permainan teka-teki dengan ukuran *grid* 4 x 4 yang belum diselesaikan.

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

└ Dasar Teori

└ Calcudoku

└ Contoh Permainan

Contoh Permainan

3	8 +	3 -	
7 +			
	8 +	8 +	
			1

Figure 1: Contoh permainan teka-teki dengan ukuran *grid* 4 x 4 yang belum diselesaikan.

Contoh Solusi

³ 3	⁸⁺ 2	³⁻ 1	4
⁷⁺ 1	4	2	3
4	⁸⁺ 1	⁸⁺ 3	2
2	3	4	¹ 1

Figure 2: Solusi untuk permainan teka-teki Calcudoku yang diberikan pada Gambar 1.

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

└ Dasar Teori

└ Calcudoku

└ Contoh Solusi

Contoh Solusi

¹	¹⁺	¹⁺	
3	2	1	4
¹⁺	1	4	2
	4	1	3
	2	3	4
			1

Figure 2: Solusi untuk permainan teka-teki Calcudoku yang diberikan pada Gambar 1.

Permasalahan Utama dalam Menyelesaikan Calcudoku

- Untuk menyelesaikan sebuah teka-teki Calcudoku, pemain harus pertama-tama memahami dua permasalahan utama dari teka-teki ini, yaitu:
 - Angka-angka mana yang harus dimasukkan ke dalam sebuah *cage*
 - Dalam urutan apa angka-angka tersebut harus dimasukkan ke dalam sebuah *cage*
- Cara yang paling mudah untuk menyelesaikan teka-teki ini adalah dengan mengeliminasi angka-angka yang sudah digunakan dan mencoba satu per satu angka yang mungkin (*trial and error*).

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

- Dasar Teori

—Calcudoku

—Permasalahan Utama dalam Menyelesaikan Calcudoku

Untuk menyelesaikan sebuah teka-teki Calcudoku, pemain harus pertama-tama memahami dua permasalahan utama dari teka-teki ini, yaitu:

1. Angka-angka mana yang harus dimasukkan ke dalam sebuah *cage*
2. Dalam urutan apa angka-angka tersebut harus dimasukkan ke dalam sebuah *cage*

Seperti kebanyakan permainan teka-teki angka, cara yang paling mudah untuk menyelesaikan teka-teki ini adalah dengan mengeliminasi angka-angka yang sudah digunakan dan mencoba satu per satu angka yang mungkin (*trial and error*).

- Untuk menyelesaikan sebuah teka-teki *Calculus*, pemain harus pertama-tama memahami dua permasalahan utama dari teka-teki ini, yaitu:
 - Angka-angka mana yang harus dimasukkan ke dalam sebuah cage
 - Dalam urutan apa angka-angka tersebut harus dimasukkan ke dalam sebuah cage
- Cara yang paling mudah untuk menyelesaikan teka-teki ini adalah dengan mengeliminasi angka-angka yang sudah digunakan dan mencoba satu per satu angka yang mungkin (*trial and error*).

Tahapan Pengisian Calcudoku

- Dalam pengisian teka-teki ini ada dua tahapan, yaitu:
- Mencari *cage* yang hanya berukuran 1 sel
- Mencari mencari *cage* yang hanya mempunyai satu kemungkinan kombinasi angka

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

- Dasar Teori

—Calcudoku

—Tahapan Pengisian Calcudoku

- Dalam pengisian teka-teki ini ada dua tahapan, yaitu:
- Mencari cage yang hanya berukuran 1 sel
- Mencari mencari cage yang hanya mempunyai satu kemungkinan kombinasi angka

Dalam pengisian teka-teki ini ada dua tahapan, yaitu:

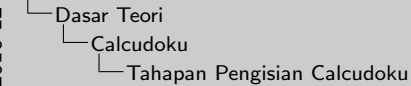
1. Mencari *cage* yang hanya berukuran 1 sel, karena *cage* ini tidak menghasilkan pertanyaan angka apa dan urutan apa. Tahap ini adalah tahap yang paling jelas. Contoh, pada Gambar 1, *cage* pada sudut kiri atas dan *cage* pada sudut kanan bawah hanya berukuran 1 sel, dan dapat langsung diisi dengan angka tujuannya.
2. Mencari mencari *cage* yang hanya mempunyai satu kemungkinan kombinasi angka, sehingga masalah angka-angka apa yang harus diisi dalam *cage* tersebut terjawab. Contoh, *cage* pada sudut kanan atas mempunyai aturan "3-", artinya angka tujuannya adalah 3 dengan menggunakan operasi pengurangan. Satu-satunya pasangan angka dari himpunan $\{1,2,3,4\}$ yang akan menghasilkan angka 3 saat satu angka dikurangkan dari angka yang lainnya adalah $\{1,4\}$. Namun masalahnya adalah urutan angka-angka yang harus dimasukkan. Dalam kasus ini, untungnya, sel pada sudut kanan bawah sudah diisi dengan angka 1, maka angka 1 tidak bisa digunakan lagi pada kolom yang paling kanan. Jadi, dengan menggunakan cara eliminasi, sel pada sudut kanan atas harus diisi dengan angka 4 dan sel di sebelah kirinya, yaitu sel pada baris yang paling atas dan kolom ketiga dari kiri, harus diisi dengan angka 1. Hal ini memberikan solusi untuk sel pada baris yang paling atas dan kolom kedua dari kiri, yaitu angka 2, karena angka 2 adalah angka yang

Tahapan Pengisian Calcudoku

- Seiring dengan meningkatnya tingkat kesulitan, langkah berikutnya tidak akan langsung muncul dengan jelas
- Kadang-kadang, pemain mencapai titik dimana langkah berikutnya tidak pasti
- Pemain harus menebak langkah-langkah berikutnya dan melihat apakah langkah ini akan menghasilkan solusinya. Jika tidak, pemain harus mundur kembali ke titik ketidakpastian tersebut.

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku



Seiring dengan meningkatnya tingkat kesulitan, langkah berikutnya tidak akan langsung muncul dengan jelas. Kadang-kadang, pemain mencapai titik dimana langkah berikutnya tidak pasti. Pemain harus menebak langkah-langkah berikutnya dan melihat apakah langkah ini akan menghasilkan solusinya. Jika tidak, pemain harus mundur kembali ke titik ketidakpastian tersebut.

- Seiring dengan meningkatnya tingkat kesulitan, langkah berikutnya tidak akan langsung muncul dengan jelas
- Kadang-kadang, pemain mencapai titik dimana langkah berikutnya tidak pasti
- Pemain harus menebak langkah-langkah berikutnya dan melihat apakah langkah ini akan menghasilkan solusinya. Jika tidak, pemain harus mundur kembali ke titik ketidakpastian tersebut.

2016-12-04

- Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku
 - Dasar Teori
 - Calcudoku
 - Mendefinisikan Permasalahan Calcudoku

- Sebuah teka-teki Caludoku dengan ukuran $n \times n$, dengan n melambangkan jumlah sel dalam satu baris atau satu kolom, mempunyai n^2 sel.
- Sel yang terletak dalam baris b dan kolom c diberi label $C_{b,c} = bn + c$.
- Nilai dari sel tersebut adalah $V(C_{b,c}) \in \{1, 2, \dots, n\}$.
- Sebuah cage, yang diberi label A , adalah sebuah himpunan dari sel, yaitu $A = \{C_{b,c}\}$.
- Setiap cage terhubung dengan satu operator aritmatika $O_i \in \{+, -, \times, \div\}$, dan satu angka tujuan $H_i \in N$.

Mendefinisikan Permasalahan Calcudoku

- 3 aturan dalam mendefinisikan masalah dalam Calcludoku adalah sebagai berikut
- $|A_i| = 1 \rightarrow O_i = \phi$, artinya setiap *cage* yang jumlah selnya 1 dengan operasi matematika yang terkait dengan *cage* tersebut bersifat homeomorfik (setara).
- $O_i \in -, \div \rightarrow |A_i| = 2$, artinya jika operasi yang digunakan dalam sebuah *cage* adalah pengurangan atau pembagian, maka jumlah sel dalam *cage* tersebut harus 2.
- $\forall C_{b,k} \rightarrow C_{b,k} \in \exists! A_i$, artinya setiap sel hanya boleh menjadi anggota dari satu dan hanya satu *cage*.

Menurut Johanna, Lukas, dan Saputra, tiga aturan dalam mendefinisikan masalah dalam Calculudoku adalah sebagai berikut:

1. $|A_i| = 1 \rightarrow O_i = \phi$, artinya setiap *cage* yang jumlah selnya 1 dengan operasi matematika yang terkait dengan *cage* tersebut bersifat homeomorfik (setara)
2. $O_i \in -, \div \rightarrow |A_i| = 2$, artinya jika operasi yang digunakan dalam sebuah *cage* adalah pengurangan atau pembagian, maka jumlah sel dalam *cage* tersebut harus 2
3. $\forall C_{b,k} \rightarrow C_{b,k} \in \exists! A_i$, artinya setiap sel hanya boleh menjadi anggota dari satu dan hanya satu *cage*

Mendefinisikan Permasalahan Calcudoku

- Tujuan dari teka-teki ini adalah untuk mencari nilai $V(C_{b,k})$ dan memenuhi persyaratan berikut
- $|A_i| = 1 \wedge C_{b,k} \in A_i \rightarrow V(C_{b,k}) = H_i$, artinya jika sel adalah bagian dari sebuah *cage* yang jumlah selnya 1, maka nilai dari sel tersebut adalah angka tujuan dari *cage* tersebut
- $O_i \in \{-\} \wedge A_i = \{C_{a,b}, C_{p,q}\} \rightarrow |V(C_{a,b}) - V(C_{p,q})| = H_i$, artinya nilai absolut dari hasil pengurangan nilai kedua sel di dalam *cage* tersebut adalah angka tujuan dari *cage* tersebut
- $O_i \in \{\div\} \wedge A_i = \{C_{a,b}, C_{p,q}\} \rightarrow V(C_{a,b})/V(C_{p,q}) = H_i$, artinya nilai dari hasil pembagian nilai kedua sel di dalam *cage* tersebut adalah angka tujuan dari *cage* tersebut
- $O_i \in \{+\} \rightarrow \sum_{C_{b,k} \in A_i} V(C_{b,k}) = H_i$, artinya nilai dari hasil penjumlahan dari nilai semua sel di dalam *cage* tersebut adalah angka tujuan dari *cage* tersebut
- $O_i \in \{\times\} \rightarrow \prod_{C_{b,k} \in A_i} V(C_{b,k}) = H_i$, artinya nilai dari hasil perkalian dari nilai semua sel di dalam *cage* tersebut adalah angka tujuan dari *cage* tersebut

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

Dasar Teori

Calcudoku

Mendefinisikan Permasalahan Calcudoku

- Tujuan dari teka-teki ini adalah untuk mencari nilai $V(C_{b,k})$ dan memenuhi persyaratan berikut
- $|A_i| = 1 \wedge C_{b,k} \in A_i \rightarrow V(C_{b,k}) = H_i$, artinya jika sel adalah bagian dari sebuah *cage* yang jumlah selnya 1, maka nilai dari sel tersebut adalah angka tujuan dari *cage* tersebut
- $O_i \in \{-\} \wedge A_i = \{C_{a,b}, C_{p,q}\} \rightarrow |V(C_{a,b}) - V(C_{p,q})| = H_i$, artinya nilai absolut dari hasil pengurangan nilai kedua sel di dalam *cage* tersebut adalah angka tujuan dari *cage* tersebut
- $O_i \in \{\div\} \wedge A_i = \{C_{a,b}, C_{p,q}\} \rightarrow V(C_{a,b})/V(C_{p,q}) = H_i$, artinya nilai dari hasil pembagian nilai kedua sel di dalam *cage* tersebut adalah angka tujuan dari *cage* tersebut
- $O_i \in \{+\} \rightarrow \sum_{C_{b,k} \in A_i} V(C_{b,k}) = H_i$, artinya nilai dari hasil penjumlahan dari nilai semua sel di dalam *cage* tersebut adalah angka tujuan dari *cage* tersebut
- $O_i \in \{\times\} \rightarrow \prod_{C_{b,k} \in A_i} V(C_{b,k}) = H_i$, artinya nilai dari hasil perkalian dari nilai semua sel di dalam *cage* tersebut adalah angka tujuan dari *cage* tersebut

Menurut Johanna, Lukas, dan Saputra, tujuan dari teka-teki ini adalah untuk mencari nilai $V(C_{b,k})$ dan memenuhi persyaratan berikut:

1. $|A_i| = 1 \wedge C_{b,k} \in A_i \rightarrow V(C_{b,k}) = H_i$, artinya jika sel adalah bagian dari sebuah *cage* yang jumlah selnya 1, maka nilai dari sel tersebut adalah angka tujuan dari *cage* tersebut.
2. $O_i \in \{-\} \wedge A_i = \{C_{a,b}, C_{p,q}\} \rightarrow |V(C_{a,b}) - V(C_{p,q})| = H_i$, artinya jika sebuah *cage* yang operasi matematikanya adalah pengurangan, maka nilai absolut dari hasil pengurangan nilai kedua sel di dalam *cage* tersebut adalah angka tujuan dari *cage* tersebut.
3. $O_i \in \{\div\} \wedge A_i = \{C_{a,b}, C_{p,q}\} \rightarrow V(C_{a,b})/V(C_{p,q}) = H_i$, artinya jika sebuah *cage* yang operasi matematikanya adalah pembagian, maka nilai dari hasil pembagian nilai kedua sel di dalam *cage* tersebut adalah angka tujuan dari *cage* tersebut.
4. $O_i \in \{+\} \rightarrow \sum_{C_{b,k} \in A_i} V(C_{b,k}) = H_i$, artinya jika sebuah *cage* yang operasi matematikanya adalah penjumlahan, maka nilai dari hasil penjumlahan dari nilai semua sel di dalam *cage* tersebut adalah angka tujuan dari *cage* tersebut.
5. $O_i \in \{\times\} \rightarrow \prod_{C_{b,k} \in A_i} V(C_{b,k}) = H_i$, artinya jika sebuah *cage* yang operasi matematikanya adalah perkalian, maka nilai dari hasil perkalian dari nilai semua sel di dalam *cage* tersebut adalah angka tujuan dari *cage* tersebut.

Algoritma *Backtracking*

- Sebuah algoritma umum yang mencari solusi dengan mencoba salah satu dari beberapa pilihan, jika pilihan yang dipilih ternyata salah, komputasi dimulai lagi pada titik pilihan dan mencoba pilihan lainnya
- Untuk bisa melacak kembali langkah-langkah yang telah dipilih, maka algoritma harus secara eksplisit menyimpan jejak dari setiap langkah yang sudah pernah dipilih, atau menggunakan rekursi (*recursion*)
- Rekursi dipilih karena jauh lebih mudah daripada harus menyimpan jejak setiap langkah yang pernah dipilih
- Hal ini menyebabkan algoritma ini biasanya berbasis DFS (*Depth First Search*)

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

└ Dasar Teori

└└ Algoritma *Backtracking*

└└└ Algoritma *Backtracking*

Algoritma *Backtracking*

- Sebuah algoritma umum yang mencari solusi dengan mencoba salah satu dari beberapa pilihan, jika pilihan yang dipilih ternyata salah, komputasi dimulai lagi pada titik pilihan dan mencoba pilihan lainnya
- Untuk bisa melacak kembali langkah-langkah yang telah dipilih, maka algoritma harus secara eksplisit menyimpan jejak dari setiap langkah yang sudah pernah dipilih, atau menggunakan rekursi (*recursion*)
- Rekursi dipilih karena jauh lebih mudah daripada harus menyimpan jejak setiap langkah yang pernah dipilih
- Hal ini menyebabkan algoritma ini biasanya berbasis DFS (*Depth First Search*)

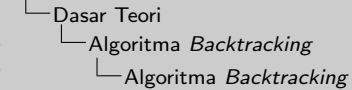
Algoritma *backtracking* adalah sebuah algoritma umum yang mencari solusi dengan mencoba salah satu dari beberapa pilihan, jika pilihan yang dipilih ternyata salah, komputasi dimulai lagi pada titik pilihan dan mencoba pilihan lainnya. Untuk bisa melacak kembali langkah-langkah yang telah dipilih, maka algoritma harus secara eksplisit menyimpan jejak dari setiap langkah yang sudah pernah dipilih, atau menggunakan rekursi (*recursion*). Rekursi dipilih karena jauh lebih mudah daripada harus menyimpan jejak setiap langkah yang pernah dipilih. Hal ini menyebabkan algoritma ini biasanya berbasis DFS (*Depth First Search*).

Algoritma *Backtracking*

- Pertama kali diperkenalkan pada tahun 1950 oleh D.H. Lehmer sebagai perbaikan algoritma *brute force*
- Algoritma ini terbukti efektif untuk menyelesaikan banyak permainan logika karena algoritma itu terutama berguna untuk menyelesaikan masalah-masalah *constraint satisfaction*, di mana sekumpulan objek harus memenuhi sejumlah batasan

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku



Algoritma *backtracking* pertama kali diperkenalkan pada tahun 1950 oleh D.H. Lehmer sebagai perbaikan algoritma *brute force*. Algoritma ini lalu dikembangkan lebih lanjut oleh R.J. Walker, S.W. Golomb, dan L.D. Baumert. Algoritma ini terbukti efektif untuk menyelesaikan banyak permainan logika (misalnya *tic tac toe*, *maze*, catur, dan lain-lain) karena algoritma itu terutama berguna untuk menyelesaikan masalah-masalah *constraint satisfaction*, di mana sekumpulan objek harus memenuhi sejumlah batasan.

- Pertama kali diperkenalkan pada tahun 1950 oleh D.H. Lehmer sebagai perbaikan algoritma brute force
- Algoritma ini terbukti efektif untuk menyelesaikan banyak permainan logika karena algoritma itu terutama berguna untuk menyelesaikan masalah-masalah constraint satisfaction, di mana sekumpulan objek harus memenuhi sejumlah batasan

Sifat-Sifat Umum Algoritma *Backtracking*

- Implementasi algoritma *backtracking* memiliki beberapa sifat umum, yaitu:
 - Ruang solusi (*solution space*)
 - Fungsi pembangkit (*generating function*)
 - Fungsi pembatas (*generating function*)

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

- Dasar Teori

—Algoritma *Backtracking*

—Sifat-Sifat Umum Algoritma *Backtracking*

Ruang Solusi

- Solusi untuk masalah ini dinyatakan sebagai sebuah vektor X dengan n -tuple:

$$X = (x_1, x_2, \dots, x_n), x_i \in S_i$$

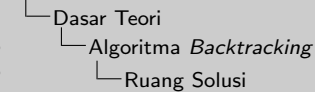
di mana adalah mungkin bahwa:

$$S_1 = S_2 = \dots = S_n$$

- n adalah jumlah sel dalam satu baris atau kolom
- X adalah sebuah *tuple* yang berukuran n^2 , yang merepresentasikan isi dari setiap sel dalam *grid*, dimulai pada sel pada sudut kiri atas, lalu bergerak ke sel di sebelah kanannya dalam baris yang sama, jika sudah mencapai sel yang paling kanan maka bergerak ke sel yang paling kiri pada baris dibawahnya, hingga berakhir di sel pada sudut kanan bawah
- S_i adalah sebuah himpunan yang berisi angka-angka dari 1 sampai n

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku



Solution space

Solusi untuk masalah ini dinyatakan sebagai sebuah vektor X dengan n -tuple:

$$X = (x_1, x_2, \dots, x_n), x_i \in S_i$$

di mana adalah mungkin bahwa:

$$S_1 = S_2 = \dots = S_n$$

n adalah jumlah sel dalam satu baris atau kolom. X adalah sebuah *tuple* yang berukuran n^2 , yang merepresentasikan isi dari setiap sel dalam *grid*, dimulai pada sel pada sudut kiri atas, lalu bergerak ke sel di sebelah kanannya dalam baris yang sama, jika sudah mencapai sel yang paling kanan maka bergerak ke sel yang paling kiri pada baris dibawahnya, hingga berakhir di sel pada sudut kanan bawah. S_i adalah sebuah himpunan yang berisi angka-angka dari 1 sampai n .

Fungsi Pembangkit

- Fungsi pembangkit X_k dinyatakan sebagai:

$$T(k)$$

di mana $T(k)$ membangkitkan nilai X_k , dari 1 sampai n , yang merupakan komponen dari vektor solusi

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

- Dasar Teori

—Algoritma *Backtracking*

—Fungsi Pembangkit

Fungsi pembangkit X_k

Fungsi pembangkit X_k dinyatakan sebagai:

$$T(k)$$

di mana $T(k)$ membangkitkan nilai X_k , dari 1 sampai n , yang merupakan komponen dari vektor solusi.

- Fungsi pembangkit X_k dinyatakan sebagai:

$$T(k)$$
 di mana $T(k)$ membangkitkan nilai X_k , dari 1 sampai n yang merupakan komponen dari vektor solusi

Fungsi Pembatas

- Fungsi pembatas dinyatakan sebagai:

$$B(x_1, x_2, \dots, x_k)$$

di mana B bernilai *true* jika (x_1, x_2, \dots, x_k) mengarah ke solusi. Jika B bernilai *true*, maka nilai $x_k + 1$ akan terus dibangkitkan, dan jika B bernilai *false*, maka (x_1, x_2, \dots, x_k) akan dibuang

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

- Dasar Teori

—Algoritma *Backtracking*

—Fungsi Pembatas

Fungsi pembangkit X_k Fungsi pembatas
Fungsi pembatas dinyatakan sebagai:

$$B(x_1, x_2, \dots, x_k)$$

di mana B bernilai *true* jika (x_1, x_2, \dots, x_k) mengarah ke solusi. Jika B bernilai *true*, maka nilai $x_k + 1$ akan terus dibangkitkan, dan jika B bernilai *false*, maka (x_1, x_2, \dots, x_k) akan dibuang.

- Fungsi pembatas dinyatakan sebagai:

$$B(x_1, x_2, \dots, x_k)$$

di mana B bernilai *true* jika (x_1, x_2, \dots, x_k) mengarah ke solusi. Jika B bernilai *true*, maka nilai $x_k + 1$ akan terus dibangkitkan, dan jika B bernilai *false*, maka (x_1, x_2, \dots, x_k) akan dibuang

Ruang Solusi

- Disusun dalam sebuah struktur berbentuk pohon (*tree*)
- Setiap simpul (*node*) merepresentasikan keadaan masalah
- Setiap sisi (*edge*) diberi label x_i
- Jalur dari akar (*root*) ke daun (*leaf*) merepresentasikan sebuah jawaban yang mungkin
- Semua jalur yang dikumpulkan bersama-sama membentuk ruang solusi
- Struktur pohon ini disebut sebagai *state space tree*

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

- Dasar Teori

—Algoritma *Backtracking*

—Ruang Solusi

Ruang solusi untuk algoritma *backtracking* disusun dalam sebuah struktur berbentuk pohon (*tree*), di mana setiap simpul (*node*) merepresentasikan keadaan masalah dan sisi (*edge*) diberi label x_i . Jalur dari akar (*root*) ke daun (*leaf*) merepresentasikan sebuah jawaban yang mungkin, dan semua jalur yang dikumpulkan bersama-sama membentuk ruang solusi. Struktur pohon ini disebut sebagai *state space tree*. Gambar 3 menggambarkan contoh sebuah *state space tree*.

- Disusun dalam sebuah struktur berbentuk pohon (tree)
- Setiap simpul (node) merepresentasikan keadaan masalah
- Setiap sisi (edge) diberi label x_i
- Jalur dari akar (root) ke daun (leaf) merepresentasikan sebuah jawaban yang mungkin
- Semua jalur yang dikumpulkan bersama-sama membentuk ruang solusi
- Struktur pohon ini disebut sebagai state space tree

Ruang Solusi

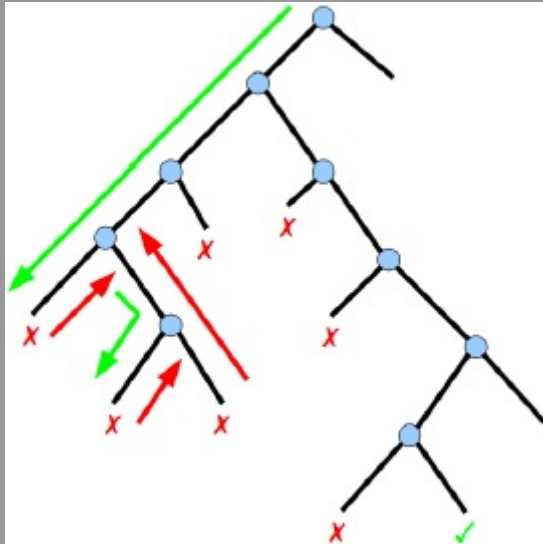


Figure 3: Ilustrasi *State space tree* yang digunakan dalam algoritma *backtracking*

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

└ Dasar Teori

└ Algoritma *Backtracking*

└ Ruang Solusi

Ruang Solusi

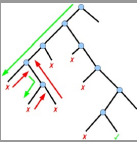


Figure 3: Ilustrasi *State space tree* yang digunakan dalam algoritma *backtracking*

Langkah-Langkah Penggunaan *State Space Tree*

- Langkah-langkah dalam menggunakan *state space tree* untuk mencari solusi adalah:
- Solusi dicari dengan membangun jalur dari akar ke daun menggunakan algoritma DFS
- Simpul yang terbentuk disebut sebagai simpul hidup (*live nodes*)
- Simpul yang sedang diperluas disebut sebagai *expand nodes* atau *E-nodes*
- Setiap kali sebuah *E-node* sedang diperluas, jalur yang dikembangkannya menjadi lebih panjang
- Jika jalur yang sedang dikembangkan tidak mengarah ke solusi, maka *E-node* dimatikan dan menjadi simpul mati (*dead node*)

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

- Dasar Teori

– Algoritma *Backtracking*

—Langkah-Langkah Penggunaan *State Space Tree*

Langkah-langkah dalam menggunakan *state space tree* untuk mencari solusi adalah [?]:

- Solusi dicari dengan membangun jalur dari akar ke daun menggunakan algoritma DFS.
- Simpul yang terbentuk disebut sebagai simpul hidup (*live nodes*).
- Simpul yang sedang diperluas disebut sebagai *expand nodes* atau *E-nodes*.
- Setiap kali sebuah *E-node* sedang diperluas, jalur yang dikembangkannya menjadi lebih panjang.
- Jika jalur yang sedang dikembangkan tidak mengarah ke solusi, maka *E-node* dimatikan dan menjadi simpul mati (*dead node*).

- Langkah-langkah dalam menggunakan *state space tree* untuk mencari solusi adalah:
 - Solusi dicari dengan membangun jalur dari akar ke daun menggunakan algoritma DFS
 - Simpul yang terbentuk disebut sebagai simpul hidup (*live nodes*)
 - Simpul yang sudah diperluas disebut sebagai *expand nodes* atau E-node
 - Setiap kali sebuah E-node sudah diperluas, jalur yang dikembangkanya menjadi lebih panjang
 - Jika jalur yang sudah dikembangkan tidak mengarah ke solusi, maka E-node dimatikan dan menjadi simpul mati (*dead node*)

2016-12-04

- └─ Dasar Teori
 - └─ Algoritma *Backtracking*
 - └─ Langkah-Langkah Penggunaan *State Space Tree*

Langkah-Langkah Penggunaan State Space Tree

- Lanjutkan dari slide sebelumnya:
- Fungsi yang digunakan untuk memuatkan E-node adalah implementasi dari fungsi pembatas
- Simpul mati tidak akan diperiksa
- Jika jalur yang sedang dibangun berakhir dengan simpul mati, proses akan mundur ke simpul sebelumnya
- Simpul sebelumnya terus mengembangkan simpul anak (child node) lainnya, yang kemudian menjadi E-node baru
- Pencarian selesai jika simpul tujuan tercapai

Kompleksitas Waktu

- Setiap simpul di dalam *state space tree* terkait dengan panggilan rekursif
- Jika jumlah simpul di dalam pohon $2n$ atau $n!$, maka pada kasus terburuk untuk algoritma *backtracking* ini memiliki kompleksitas waktu $O(p(n)2n)$ atau $O(q(n)n!)$, dengan $p(n)$ dan $q(n)$ sebagai polinomial dengan n -derajat menyatakan waktu komputasi untuk setiap simpul

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

- Dasar Teori

– Algoritma *Backtracking*

- Kompleksitas Waktu

Setiap simpul di dalam *state space tree* terkait dengan panggilan rekursif. Jika jumlah simpul di dalam pohon $2n$ atau $n!$, maka pada kasus terburuk untuk algoritma *backtracking* ini memiliki kompleksitas waktu $O(p(n)2n)$ atau $O(q(n)n!)$, dengan $p(n)$ dan $q(n)$ sebagai polinomial dengan n -derajat menyatakan waktu komputasi untuk setiap simpul.

- Setiap simpul di dalam state space tree terkait dengan panggilan rekursif
- Jika jumlah simpul di dalam pohon $2n$ atau $n!$, maka pada kasus terburuk untuk algoritma backtracking ini memiliki kompleksitas waktu $O(p(n)2n)$ atau $O(q(n)n!)$, dengan $p(n)$ dan $q(n)$ sebagai polinomial dengan n -derajat menyatakan waktu komputasi untuk setiap simpul

Ruang Solusi

- Ruang solusi untuk sebuah permainan teka-teki Calcutoku dengan *grid* yang berukuran $n \times n$ adalah $X = (x_1, x_2, \dots, x_m), x_i \in \{1, 2, \dots, n\}$, dengan $m = n^2$
- Fungsi pembangkit membangkitkan sebuah integer secara berurutan dari 1 sampai n sebagai x_k
- Fungsi pembatas menggabungkan tiga fungsi pemeriksa pembatas (*constraint checking*), yaitu:
 - Fungsi pemeriksa kolom (*column checking*)
 - Fungsi pemeriksa baris (*row checking*)
 - Fungsi pemeriksa *grid* (*grid checking*)

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcutoku

Dasar Teori

Algoritma *Backtracking*

Ruang Solusi

Ruang solusi untuk sebuah permainan teka-teki Calcutoku dengan *grid* yang berukuran $n \times n$ adalah $X = (x_1, x_2, \dots, x_m), x_i \in \{1, 2, \dots, n\}$, dengan $m = n^2$. Fungsi pembangkit membangkitkan sebuah integer secara berurutan dari 1 sampai n sebagai x_k . Fungsi pembatas menggabungkan tiga fungsi pemeriksa pembatas (*constraint checking*), yaitu fungsi pemeriksa kolom (*column checking*), fungsi pemeriksa baris (*row checking*), dan fungsi pemeriksa *grid* (*grid checking*).

Ruang Solusi

- Ruang solusi untuk sebuah permainan teka-teki Calcutoku dengan *grid* yang berukuran $n \times n$ adalah $X = (x_1, x_2, \dots, x_m), x_i \in \{1, 2, \dots, n\}$, dengan $m = n^2$
- Fungsi pembangkit membangkitkan sebuah integer secara berurutan dari 1 sampai n sebagai x_k
- Fungsi pembatas menggabungkan tiga fungsi pemeriksa pembatas (*constraint checking*), yaitu:
 - Fungsi pemeriksa kolom (*column checking*)
 - Fungsi pemeriksa baris (*row checking*)
 - Fungsi pemeriksa *grid* (*grid checking*)

Fungsi Pembatas

- Fungsi pemeriksa kolom menghasilkan nilai *true* jika x_k belum ada di dalam kolom dan menghasilkan nilai *false* jika sebaliknya
- Fungsi pemeriksa baris menghasilkan nilai *true* jika x_k belum ada di dalam baris dan menghasilkan nilai *false* jika sebaliknya
- Fungsi pemeriksa *grid* memeriksa operator pada *grid* dan memeriksa berdasarkan operator yang telah ditentukan

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

– Dasar Teori

—Algoritma *Backtracking*

—Fungsi Pembatas

Fungsi pemeriksa kolom menghasilkan nilai *true* jika x_k belum ada di dalam kolom dan menghasilkan nilai *false* jika x_k sudah ada di dalam kolom.

Fungsi pemeriksa baris menghasilkan nilai *true* jika x_k belum ada di dalam baris dan menghasilkan nilai *false* jika x_k sudah ada di dalam baris.

Fungsi memeriksa *grid* memeriksa operator pada *grid* dan memeriksa berdasarkan operator yang telah ditentukan.

- Fungsi pemeriksa kolom menghasilkan nilai $true$ jika x_k belum ada di dalam kolom dan menghasilkan nilai $false$ jika sebaliknya
- Fungsi pemeriksa baris menghasilkan nilai $true$ jika x_k belum ada di dalam baris dan menghasilkan nilai $false$ jika sebaliknya
- Fungsi pemeriksa *grid* memeriksa operator pada *grid* dan memeriksa berdasarkan operator yang telah ditentukan

Operator-Operator untuk Fungsi Pemeriksa *Grid*

- Ada 5 operator yang digunakan dalam fungsi ini, yaitu:
- Operator penjumlahan (+), fungsi menghasilkan nilai *true* jika hasil penjumlahan semua nilai yang ada pada *grid* ditambah dengan x_k kurang dari atau sama dengan nilai tujuan, dan menghasilkan nilai *false* jika sebaliknya
- Operator pengurangan (-), fungsi menghasilkan nilai *true* jika kedua sel dalam *grid* kosong, atau jika ada satu sel yang kosong dan hasil dari x_k dikurangi dengan nilai dari sel yang lainnya atau hasil dari nilai dari sel yang lainnya dikurangi dengan x_k menghasilkan nilai tujuan, dan menghasilkan nilai *false* jika sebaliknya

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

└ Dasar Teori

└ Algoritma *Backtracking*

└ Operator-Operator untuk Fungsi Pemeriksa *Grid*

Ada 5 operator yang digunakan dalam fungsi ini, yaitu:

- Operator penjumlahan (+), fungsi menghasilkan nilai *true* jika hasil penjumlahan semua nilai yang ada pada *grid* ditambah dengan x_k kurang dari atau sama dengan nilai tujuan, dan menghasilkan nilai *false* jika jumlah semua nilai yang ada pada *grid* ditambah x_k lebih dari nilai tujuan.
- Operator pengurangan (-), fungsi menghasilkan nilai *true* jika kedua sel dalam *grid* kosong, atau jika ada satu sel yang kosong dan hasil dari x_k dikurangi dengan nilai dari sel yang lainnya atau hasil dari nilai dari sel yang lainnya dikurangi dengan x_k menghasilkan nilai tujuan, dan menghasilkan nilai *false* jika ada satu sel kosong dan hasil dari x_k dikurangi dengan nilai dari sel yang lainnya atau hasil dari nilai dari sel yang lainnya dikurangi dengan x_k tidak menghasilkan nilai tujuan.

Operator-Operator untuk Fungsi Pemeriksa *Grid*

- Ada 5 operator yang digunakan dalam fungsi ini, yaitu:
- Operator penjumlahan (+), fungsi menghasilkan nilai *true* jika hasil penjumlahan semua nilai yang ada pada *grid* ditambah dengan x_k kurang dari atau sama dengan nilai tujuan, dan menghasilkan nilai *false* jika sebaliknya
- Operator pengurangan (-), fungsi menghasilkan nilai *true* jika kedua sel dalam *grid* kosong, atau jika ada satu sel yang kosong dan hasil dari x_k dikurangi dengan nilai dari sel yang lainnya atau hasil dari nilai dari sel yang lainnya dikurangi dengan x_k menghasilkan nilai tujuan, dan menghasilkan nilai *false* jika sebaliknya

2016-12-04

- └─ Dasar Teori
 - └─ Algoritma *Backtracking*
 - └─ Operator-Operator untuk Fungsi Pemeriksa *Grid*

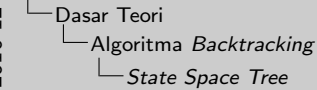
- Operator perkalian (\times), fungsi menghasilkan nilai true jika hasil perkalian dari semua nilai x ada pada grid dikali dengan x kurang dari atau sama dengan nilai tujuan, dan menghasilkan nilai false jika sebaliknya.
- Operator pembagian (\div), fungsi menghasilkan nilai true jika kedua sel dalam grid kosong, atau jika ada satu sel yang kosong dan hasil dari x dibagi dengan dari nilai sel yang lainnya atau hasil dari nilai dari sel yang lainnya dibagi dengan x menghasilkan nilai tujuan, dan menghasilkan nilai false jika sebaliknya.
- Operator pengurangan ($-$), fungsi menghasilkan nilai true jika x sama dengan nilai tujuan, dan menghasilkan nilai false jika sebaliknya.

State Space Tree

- *State space tree* bersifat dinamis, berkembang secara terus-menerus sampai solusi ditemukan
- Tinggi pohon yang dikembangkan untuk menyelesaikan sebuah teka-teki dengan ukuran $n \times n$ seharusnya memiliki tinggi $n^2 + 1$ saat mencapai simpul tujuannya, dengan jalur dari simpul akar ke simpul tujuan merepresentasikan semua angka yang digunakan untuk mengisi *grid* dari sel pada sudut kiri atas ke sel pada sudut kanan bawah

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku



State space tree bersifat dinamis, berkembang secara terus-menerus sampai solusi ditemukan. Tinggi pohon yang dikembangkan untuk menyelesaikan sebuah teka-teki dengan ukuran $n \times n$ seharusnya memiliki tinggi $n^2 + 1$ saat mencapai simpul tujuannya, dengan jalur dari simpul akar ke simpul tujuan merepresentasikan semua angka yang digunakan untuk mengisi *grid* dari sel pada sudut kiri atas ke sel pada sudut kanan bawah.

- State space tree bersifat dinamis, berkembang secara terus-menerus sampai solusi ditemukan
- Tinggi pohon yang dikembangkan untuk menyelesaikan sebuah teka-teki dengan ukuran $n \times n$ seharusnya memiliki tinggi $n^2 - 1$ saat mencapai simpul tujuannya dengan jalur dari simpul akar ke simpul tujuan
- merepresentasikan semua angka yang digunakan untuk mengisi grid dari sel pada sudut kiri atas ke sel pada sudut kanan bawah

Cara Kerja Algoritma *Backtracking* Secara Singkat

- Singkatnya, langkah-langkah dasar dari implementasi algoritma *backtracking* dapat dijelaskan sebagai berikut:

- 1 Carilah sel pertama atau sel yang kosong di dalam *grid*
- 2 Isilah sel dengan sebuah angka dimulai dari 1 sampai n sampai sebuah angka yang berlaku (*valid*) ditemukan atau sampai angka sudah melebihi n
- 3 Jika angka untuk sel berlaku, ulangi langkah 1 dan 2
- 4 Jika angka untuk sel sudah melebihi n dan tidak ada angka dari 1 sampai n yang berlaku untuk sel tersebut, mundur ke sel sebelumnya dan cobalah kemungkinan angka berikutnya yang berlaku untuk sel tersebut
- 5 Jika tidak ada lagi sel yang kosong, solusi sudah ditemukan

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

└ Dasar Teori

└ Algoritma *Backtracking*

└ Cara Kerja Algoritma *Backtracking* Secara Singkat

Singkatnya, langkah-langkah dasar dari implementasi algoritma *backtracking* dapat dijelaskan sebagai berikut [?]:

1. Carilah sel pertama atau sel yang kosong di dalam *grid*.
2. Isilah sel dengan sebuah angka dimulai dari 1 sampai n sampai sebuah angka yang berlaku (*valid*) ditemukan atau sampai angka sudah melebihi n .
3. Jika angka untuk sel berlaku, ulangi langkah 1 dan 2.
4. Jika angka untuk sel sudah melebihi n dan tidak ada angka dari 1 sampai n yang berlaku untuk sel tersebut, mundur ke sel sebelumnya dan cobalah kemungkinan angka berikutnya yang berlaku untuk sel tersebut.
5. Jika tidak ada lagi sel yang kosong, solusi sudah ditemukan.

Cara Kerja Algoritma *Backtracking* Secara Singkat

- Singkatnya, langkah-langkah dasar dari implementasi algoritma *backtracking* dapat dijelaskan sebagai berikut:
 - 1 Carilah sel pertama atau sel yang kosong di dalam *grid*
 - 2 Isilah sel dengan sebuah angka dimulai dari 1 sampai n sampai sebuah angka yang berlaku (*valid*) ditemukan atau sampai angka sudah melebihi n
 - 3 Jika angka untuk sel berlaku, ulangi langkah 1 dan 2
 - 4 Jika angka untuk sel sudah melebihi n dan tidak ada angka dari 1 sampai n yang berlaku untuk sel tersebut, mundur ke sel sebelumnya dan cobalah kemungkinan angka berikutnya yang berlaku untuk sel tersebut
 - 5 Jika tidak ada lagi sel yang kosong, solusi sudah ditemukan

Algoritma *Hybrid Genetic*

- Dalam kasus ini, algoritma *hybrid genetic* adalah gabungan dari algoritma *rule based* dan algoritma genetik
- Algoritma *rule based* akan dijalankan sampai pada titik dimana algoritma tidak bisa menyelesaikan permainan teka-teki Calcudoku
- Jika algoritma sudah tidak bisa menyelesaikan permainan, maka algoritma genetik akan mulai dijalankan

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

- Dasar Teori

– Algoritma *Hybrid Genetic*

—Algoritma *Hybrid Genetic*

Dalam kasus ini, algoritma ini gabungan dari algoritma *rule based* dan algoritma genetik. Algoritma *rule based* akan dijalankan sampai pada titik dimana algoritma tidak bisa menyelesaikan permainan teka-teki Calcudoku. Jika algoritma sudah tidak bisa menyelesaikan permainan, maka algoritma genetik akan mulai dijalankan.

- Dalam kasus ini, algoritma *hybrid genetic* adalah gabungan dari algoritma *rule based* dan algoritma genetik
- Algoritma *rule based* akan dijalankan sampai pada titik dimana algoritma tidak bisa menyelesaikan permainan teka-teki *Calculus*
- Jika algoritma sudah tidak bisa menyelesaikan permainan, maka algoritma genetik akan mulai dijalankan

Algoritma *Rule Based*

- Sebuah algoritma berbasis aturan logika untuk menyelesaikan teka-teki Sudoku dan variasinya, termasuk Calcudoku
- Beberapa aturan logika yang digunakan dalam algoritma ini adalah:
 - *Single square rule*
 - *Naked subset rule*
 - *Hidden single rule*
 - *Evil twin rule*
 - *Killer combination*
 - *X-wing*

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

- Dasar Teori

—Algoritma *Hybrid Genetic*

- Algoritma *Rule Based*

Algoritma *rule based* adalah sebuah algoritma berbasis aturan logika untuk menyelesaikan teka-teki Sudoku dan variasinya, termasuk Calcudoku. Menurut Johanna, Lukas, dan Saputra, beberapa aturan logika yang digunakan dalam algoritma ini adalah *single square rule*, *naked subset rule*, *hidden single rule*, *evil twin rule*, *killer combination*, dan *X-wing* [?].

- Sebuah algoritma berbasis aturan logika untuk menyelesaikan teka-teki Sudoku dan variasinya, termasuk Caludoku
- Beberapa aturan logika yang digunakan dalam algoritma ini adalah:
 - Single square rule
 - Naked subset rule
 - Hidden single rule
 - Evil twin rule
 - Killer combination
 - X-wing

Aturan *Single Square*

- Digunakan jika sebuah cage hanya berisi satu sel
- Nilai dari sel tersebut sama dengan angka tujuan yang telah ditentukan

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

- Dasar Teori

—Algoritma *Hybrid Genetic*

Aturan *Single Square*

- Digunakan jika sebuah cage hanya berisi satu sel
- Nilai dari sel tersebut sama dengan angka tujuan yang telah ditentukan

Aturan *single square* digunakan jika sebuah *cage* hanya berisi satu sel. Hal ini berarti nilai dari sel tersebut sama dengan angka tujuan yang telah ditentukan.

Aturan *Naked Subset*

- Digunakan jika ada n sel dalam kolom atau baris yang sama yang mempunyai n kemungkinan nilai yang sama persis untuk mengisinya, dengan $n \geq 2$.
- Sel-sel lainnya dalam baris dan kolom tersebut tidak mungkin diisi dengan nilai yang sama dengan nilai milik n sel tersebut.

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

- Dasar Teori

—Algoritma *Hybrid Genetic*

- Aturan *Naked Subset*

Aturan *naked subset* digunakan jika ada n sel dalam kolom atau baris yang sama yang mempunyai n kemungkinan nilai yang sama persis untuk mengisinya, dengan $n \geq 2$. Hal ini berarti sel-sel lainnya dalam baris dan kolom tersebut tidak mungkin diisi dengan nilai yang sama dengan nilai milik n sel tersebut.

- Digunakan jika ada n sel dalam kolom atau baris yang sama yang mempunyai n kemungkinan nilai yang sama persis untuk mengisinya, dengan $n \geq 2$.
- Sel-sel lainnya dalam baris dan kolom tersebut tidak mungkin diisi dengan nilai yang sama dengan nilai n sel tersebut.

Aturan *Naked Subset*

3	9	4	17	6	17	18	27	5
---	---	---	----	---	----	----	----	---

Figure 4: Contoh bagaimana cara mendeteksi aturan *naked pair*

- Gambar 4 menunjukkan bagaimana cara kerja aturan ini
- Sel-sel pada kolom ke-4 dan ke-6 mempunyai tepat dua kemungkinan nilai (1 atau 7)
- Ini disebut sebagai *naked pair*
- Karena angka 1 dan 7 harus diisi pada sel-sel pada kolom ke-4 dan ke-6, maka angka 1 dan 7 bisa dieliminasi dari sel-sel pada kolom ke-7 dan ke-8

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

- Dasar Teori

—Algoritma *Hybrid Genetic*

- Aturan *Naked Subset*

3	9	4	17	6	17	18	27	5
---	---	---	----	---	----	----	----	---

Figure 4: Contoh bagaimana cara mendeteksi aturan *nake*
pair

- Gambar 4 menunjukkan bagaimana cara kerja aturan ini
- Sel-sel pada kolom ke-4 dan ke-6 mempunyai tepat dua kemungkinan nilai (1 atau 7)
- Ini disebut sebagai *naked pair*
- Karena angka 1 dan 7 harus diisi pada sel-sel pada kolom ke-4 dan ke-6, maka angka 1 dan 7 bisa dieliminasi dari sel-sel pada kolom ke-7 dan ke-8

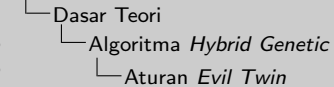
Gambar 4 menunjukkan bagaimana cara kerja aturan ini. Sel-sel pada kolom ke-4 dan ke-6 mempunyai tepat dua kemungkinan nilai (1 atau 7). Ini disebut sebagai *naked pair*. Karena angka 1 dan 7 harus diisi pada sel-sel pada kolom ke-4 dan ke-6, maka angka 1 dan 7 bisa dieliminasi dari sel-sel pada kolom ke-7 dan ke-8.

Aturan *Evil Twin*

- Digunakan jika sebuah *cage* berisikan dua sel, dan salah satu dari kedua sel sudah terisi, maka sel yang satunya lagi diisi dengan angka yang jika kedua angka dihitung dengan operasi matematika yang ditentukan maka akan menghasilkan angka tujuan yang ditentukan
- Bisa digeneralisasikan untuk *cage* yang berukuran lebih dari 2 sel
- Sel yang belum terisi yang terakhir dalam sebuah area diisi oleh sebuah nilai yang diperlukan untuk mencapai nilai tujuan menggunakan operasi matematika yang telah ditentukan.

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku



Aturan *evil twin* digunakan jika sebuah *cage* berisikan dua sel, dan salah satu dari kedua sel sudah terisi, maka sel yang satunya lagi diisi dengan angka yang jika kedua angka dihitung dengan operasi matematika yang ditentukan maka akan menghasilkan angka tujuan yang ditentukan. Aturan ini adalah aturan yang paling mudah. Kenyataannya, aturan ini bisa digeneralisasikan untuk *cage* yang berukuran lebih dari 2 sel. Sel yang belum terisi yang terakhir dalam sebuah area diisi oleh sebuah nilai yang diperlukan untuk mencapai nilai tujuan menggunakan operasi matematika yang telah ditentukan.

- Digunakan jika sebuah cage berisikan dua sel, dan salah satu dari kedua sel sudah terisi, maka sel yang satunya lagi diisi dengan angka yang jika kedua angka dihitungkan dengan operasi matematika yang ditentukan maka akan menghasilkan angka tujuan yang ditentukan
- Bisa digeneralisasikan untuk cage yang berukuran lebih dari 2 sel
- Sel yang belum terisi yang terakhir dalam sebuah area diisi oleh sebuah nilai yang diperlukan untuk mencapai nilai tujuan menggunakan operasi matematika yang telah ditentukan.

Aturan *Evil Twin*



Figure 5: Contoh aturan *evil twin*

- Contohnya, pada Gambar 5, begitu sel di sudut kiri bawah diisi oleh angka 4, maka sel di atasnya harus diisi oleh angka 9.

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

└ Dasar Teori

└ Algoritma *Hybrid Genetic*

└ Aturan *Evil Twin*

Aturan *Evil Twin*



Figure 5: Contoh aturan *evil twin*

- Contohnya, pada Gambar 5, begitu sel di sudut kiri bawah diisi oleh angka 4, maka sel di atasnya harus diisi oleh angka 9.

Contohnya, pada Gambar 5, begitu sel di sudut kiri bawah diisi oleh angka 4, maka sel di atasnya harus diisi oleh angka 9.

Aturan *Hidden Single*



Figure 6: Contoh aturan *hidden single*

- Digunakan jika sebuah angka hanya bisa diisikan dalam satu sel dalam sebuah baris atau kolom.
- Pada Gambar 6, nilai-nilai yang mungkin untuk sel yang paling kiri adalah 3, 5, dan 7
- Tetapi dalam baris ini, angka 7 harus muncul dalam salah satu selnya, dan hanya sel yang paling kiri tersebut yang memiliki kemungkinan nilai 7
- Ini disebut sebagai *hidden single*
- Sel tersebut harus diisi dengan angka 7.

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

└ Dasar Teori

└ Algoritma *Hybrid Genetic*

└ Aturan *Hidden Single*

Aturan Hidden Single

6	4	2	8	9
---	---	---	---	---

Figure 6: Contoh aturan *hidden single*

- Digunakan jika sebuah angka hanya bisa diisikan dalam satu sel dalam sebuah baris atau kolom.
- Pada Gambar 6, nilai-nilai yang mungkin untuk sel yang paling kiri adalah 3, 5, dan 7
- Tetapi dalam baris ini, angka 7 harus muncul dalam salah satu selnya, dan hanya sel yang paling kiri tersebut yang memiliki kemungkinan nilai 7
- Ini disebut sebagai *hidden single*
- Sel tersebut harus diisi dengan angka 7.

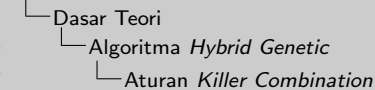
Aturan *hidden single* digunakan jika sebuah angka hanya bisa diisikan dalam satu sel dalam sebuah baris atau kolom. Aturan ini secara konsep cukup mudah, tetapi kadang-kadang sulit untuk diamati. Pada Gambar 6, nilai-nilai yang mungkin untuk sel yang paling kiri adalah 3, 5, dan 7, tetapi dalam baris ini, angka 7 harus muncul dalam salah satu selnya, dan hanya sel yang paling kiri tersebut yang memiliki kemungkinan nilai 7. Ini disebut sebagai *hidden single*. Sel tersebut harus diisi dengan angka 7.

Aturan *Killer Combination*

- Digunakan jika sebuah *cage* berisikan sel-sel yang berada dalam baris atau kolom yang sama dan operasi yang ditentukan adalah penjumlahan
- Kemungkinan angka yang unik untuk aturan *killer combination* berhubungan dengan ukuran *cage*
- Contoh, jika sebuah *cage* memiliki dua sel dan angka tujuannya adalah 3, maka kemungkinan angka yang bisa diisikan ke dalam kedua sel tersebut adalah 1 atau 2
- Hal ini berarti semua angka lainnya tidak mungkin diisikan ke dalam kedua sel tersebut
- Jika sebuah *cage* memiliki tiga sel dan angka tujuannya adalah 24, maka kemungkinan angka yang bisa diisikan ke dalam ketiga sel tersebut adalah 7, 8, atau 9

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku



Aturan *killer combination* adalah aturan yang paling krusial. Aturan ini digunakan jika sebuah *cage* berisikan sel-sel yang berada dalam baris atau kolom yang sama dan operasi yang ditentukan adalah penjumlahan. Kemungkinan angka yang unik untuk aturan *killer combination* berhubungan dengan ukuran *cage*. Contoh, jika sebuah *cage* memiliki dua sel dan angka tujuannya adalah 3, maka kemungkinan angka yang bisa diisikan ke dalam kedua sel tersebut adalah 1 atau 2. Hal ini berarti semua angka lainnya tidak mungkin diisikan ke dalam kedua sel tersebut. Contoh lain, jika sebuah *cage* memiliki tiga sel dan angka tujuannya adalah 24, maka kemungkinan angka yang bisa diisikan ke dalam ketiga sel tersebut adalah 7, 8, atau 9.

- Digunakan jika sebuah cara berisikan sel-sel yang berada dalam baris atau kolom yang sama dan operasi yang ditentukan adalah penjumlahan
- Kemungkinan angka yang unik untuk aturan killer combination berhubungan dengan ukuran cage
- Contoh, jika sebuah cage memiliki dua sel dan angka tujuannya adalah 3, maka kemungkinan angka yang bisa diisikan ke dalam cage tersebut adalah 1 atau 2
- Hal ini berarti semua angka lainnya tidak mungkin diisikan ke dalam cage tersebut
- Jika sebuah cage memiliki tiga sel dan angka tujuannya adalah 24, maka kemungkinan angka yang bisa diisikan ke dalam cage tersebut adalah 7, 8 atau 9

Aturan *Killer Combination*

Cage size	Cage value	Combination
2	3	1/2
2	4	1/3
2	17	8/9
2	16	7/9

Figure 7: Contoh aturan *killer combination* untuk cage dengan ukuran 2 sel

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

- └ Dasar Teori
 - └ Algoritma *Hybrid Genetic*
 - └ Aturan *Killer Combination*

Aturan *Killer Combination*

Cage size	Cage value	Combination
2	3	1/2
2	4	1/3
2	17	8/9
2	16	7/9

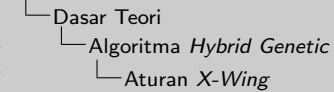
Figure 7: Contoh aturan *killer combination* untuk cage dengan ukuran 2 sel

Aturan *X-Wing*

- Digunakan jika hanya ada dua kemungkinan angka yang bisa diisikan ke dalam dua sel yang berada di dalam dua baris yang berbeda, dan dua kemungkinan angka tersebut juga berada di dalam kolom yang sama maka sel-sel lainnya dalam kolom tersebut tidak mungkin diisi oleh dua kemungkinan angka tersebut
- Juga digunakan jika hanya ada dua kemungkinan angka yang bisa diisikan ke dalam dua sel yang berada di dalam dua kolom yang berbeda, dan dua kemungkinan angka tersebut juga berada di dalam baris yang sama maka sel-sel lainnya dalam baris tersebut tidak mungkin diisi oleh dua kemungkinan angka tersebut

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku



Aturan *X-wing* digunakan jika hanya ada dua kemungkinan angka yang bisa diisikan ke dalam dua sel yang berada di dalam dua baris yang berbeda, dan dua kemungkinan angka tersebut juga berada di dalam kolom yang sama maka sel-sel lainnya dalam kolom tersebut tidak mungkin diisi oleh dua kemungkinan angka tersebut, atau jika hanya ada dua kemungkinan angka yang bisa diisikan ke dalam dua sel yang berada di dalam dua kolom yang berbeda, dan dua kemungkinan angka tersebut juga berada di dalam baris yang sama maka sel-sel lainnya dalam baris tersebut tidak mungkin diisi oleh dua kemungkinan angka tersebut.

- Digunakan jika hanya ada dua kemungkinan angka yang bisa diisikan ke dalam dua sel yang berada di dalam dua baris yang berbeda, dan dua kemungkinan angka tersebut juga berada di dalam kolom yang sama maka sel-sel lainnya dalam kolom tersebut tidak mungkin diisi oleh dua kemungkinan angka tersebut
- Juga digunakan jika hanya ada dua kemungkinan angka yang bisa diisikan ke dalam dua sel yang berada di dalam dua kolom yang berbeda, dan dua kemungkinan angka tersebut juga berada di dalam baris yang sama maka sel-sel lainnya dalam baris tersebut tidak mungkin diisi oleh dua kemungkinan angka tersebut

Aturan X-Wing



Figure 8: Contoh aturan X-wing [?]

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

└ Dasar Teori

└ Algoritma *Hybrid Genetic*

└ Aturan X-Wing

Aturan X-Wing



Figure 8: Contoh aturan X-wing [7]

Aturan *X-Wing*

- Gambar 8 menampilkan contoh penggunaan aturan *X-wing*
- Misalnya, jika sel A diisi oleh angka 7, maka angka 7 akan dieliminasi dari sel B dan sel C
- Karena sel A dengan sel C dan sel D 'terkunci', maka sel D harus diisi oleh angka 7
- Jadi, angka 7 harus di isi pada sel A dan sel D atau pada sel B dan sel C
- Angka 7 bisa dieliminasi dari sel-sel yang berwarna hijau

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

- Dasar Teori

—Algoritma *Hybrid Genetic*

– Aturan *X-Wing*

Gambar 8 menampilkan contoh penggunaan aturan *X-wing*. Misalnya, jika sel A diisi oleh angka 7, maka angka 7 akan dieliminasi dari sel B dan sel C. Karena sel A dengan sel C dan sel D 'terkunci', maka sel D harus diisi oleh angka 7. Jadi, angka 7 harus diisi pada sel A dan sel D atau pada sel B dan sel C. Angka 7 bisa dieliminasi dari sel-sel yang berwarna hijau.

- Gambar 8 menampilkan contoh penggunaan aturan X-wing.
- Misalnya, jika sel A diisi oleh angka 7, maka angka 7 akan dieliminasi dari sel B dan sel C
- Karena sel A dengan sel C dan sel D 'terkunci', maka sel D harus diisi oleh angka 7
- Jadi, angka 7 harus di isi pada sel A dan sel D atau pada sel B dan sel C
- Angka 7 bisa dieliminasi dari sel-sel yang berwarna hijau

Heuristik

- Semacam aturan tidak tertulis yang mungkin menghasilkan solusi
- Kadang-kadang efektif, tetapi tidak dijamin akan berhasil dalam setiap kasus
- Memerankan peran penting dalam strategi pencarian karena sifat eksponensial dari kebanyakan masalah
- Membantu mengurangi jumlah alternatif solusi dari angka yang bersifat eksponensial menjadi angka yang bersifat polinomial

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

- Dasar Teori

—Algoritma *Hybrid Genetic*

- Heuristik

- Semacam aturan tidak tertulis yang mungkin menghasilkan solusi
- Kadang-kadang efektif, tetapi tidak dijamin akan berhasil dalam setiap kasus
- Memerankan peran penting dalam strategi pencarian karena sifat eksponensial dari kebanyakan masalah
- Membantu mengurangi jumlah alternatif solusi dari angka yang bersifat eksponensial menjadi angka yang bersifat polinomial

Heuristik adalah semacam aturan tidak tertulis yang mungkin menghasilkan solusi. Heuristik kadang-kadang efektif, tetapi tidak dijamin akan berhasil. dalam setiap kasus. Heuristik memerankan peran penting dalam strategi pencarian karena sifat eksponensial dari kebanyakan masalah. Heuristik membantu mengurangi jumlah alternatif solusi dari angka yang bersifat eksponensial menjadi angka yang bersifat polinomial.

Pencarian Heuristik

- Sebuah teknik pencarian kecerdasan buatan (*artificial intelligence*) yang menggunakan heuristik dalam langkah-langkahnya
- Contoh teknik pencarian heuristik adalah:
 - *Generate and Test*
 - *Hill Climbing*
 - *Best First Search*

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

- Dasar Teori

—Algoritma *Hybrid Genetic*

—Pencarian Heuristik

Pencarian heuristik adalah sebuah teknik pencarian kecerdasan buatan (*artificial intelligence*) yang menggunakan heuristik dalam langkah-langkahnya. Contoh teknik pencarian heuristik adalah *Generate and Test*, *Hill Climbing*, dan *Best First Search*.

- Sebuah teknik pencarian kecerdasan buatan (*artificial intelligence*) yang menggunakan heuristik dalam langkah-langkahnya
- Contoh teknik pencarian heuristik adalah:
 - *Generate and Test*
 - *Hill Climbing*
 - *Best First Search*

Algoritma Genetik

- Salah satu teknik heuristik *Generate and Test* yang terinspirasi oleh sistem seleksi alam
- Perpaduan dari bidang biologi dan ilmu komputer.
- Algoritma ini memanipulasi informasi, biasanya disebut sebagai kromosom.

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

- Dasar Teori

—Algoritma *Hybrid Genetic*

- Algoritma Genetik

- Salah satu teknik heuristik *Generate and Test* yang terinspirasi oleh sistem seleksi alam
- Perpaduan dari bidang biologi dan ilmu komputer.
- Algoritma ini memanipulasi informasi, biasanya disebut sebagai kromosom.

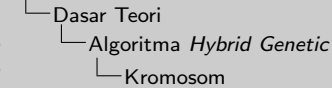
Algoritma genetik adalah salah satu teknik heuristik *Generate and Test* yang terinspirasi oleh sistem seleksi alam. Algoritma ini adalah perpaduan dari bidang biologi dan ilmu komputer. Algoritma ini memanipulasi informasi, biasanya disebut sebagai kromosom.

Kromosom

- Meng-*encode* kemungkinan jawaban untuk sebuah masalah yang diberikan.
- Dievaluasi dan diberi *fitness value* berdasarkan seberapa baikkah kromosom dalam menyelesaikan masalah yang diberikan berdasarkan kriteria yang ditentukan
- Nilai kelayakan ini digunakan sebagai probabilitas keberuntungan hidup kromosom dalam satu siklus reproduksi
- *Child chromosome* diproduksi dengan menggabungkan dua (atau lebih) *parent chromosome*
- Proses ini dirancang untuk menghasilkan kromosom-kromosom keturunan yang lebih layak
- Kromosom-kromosom ini meng-*encode* jawaban yang lebih baik, sampai solusi yang baik dan yang bisa diterima ditemukan.

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku



Kromosom ini meng-*encode* kemungkinan jawaban untuk sebuah masalah yang diberikan. Kromosom dievaluasi dan diberi *fitness value* berdasarkan seberapa baiklah kromosom dalam menyelesaikan masalah yang diberikan berdasarkan kriteria yang ditentukan oleh pembuat program. Nilai kelayakan ini digunakan sebagai probabilitas keberuntungan hidup kromosom dalam satu siklus reproduksi. Kromosom baru (kromosom anak, *child chromosome*) diproduksi dengan menggabungkan dua (atau lebih) kromosom orang tua (*parent chromosome*). Proses ini dirancang untuk menghasilkan kromosom-kromosom keturunan yang lebih layak, kromosom-kromosom ini meng-*encode* jawaban yang lebih baik, sampai solusi yang baik dan yang bisa diterima ditemukan.

- Meng-encode kemungkinan jawaban untuk sebuah masalah yang diberikan.
- Dinikmati dan diberi fitness value berdasarkan seberapa baik kromosom dalam menyelesaikan masalah yang diberikan berdasarkan kriteria yang ditentukan
- Nilai kelangkaan ini digunakan sebagai probabilitas keberuntungan hidup kromosom dalam satu siklus reproduksi
- Child chromosome diproduksi dengan menggabungkan dua (atau lebih) parent chromosome
- Proses ini dirancang untuk menghasilkan kromosom-kromosom keturunan yang lebih layak
- Kromosom-kromosom ini meng-encode jawaban yang lebih baik, sampai solusi yang baik dan yang bisa diterima ditemukan.

Cara Kerja Algoritma Genetik

- Cara kerja algoritma genetik adalah sebagai berikut:
 - 1 Menentukan populasi kromosom kemungkinan jawaban awal
 - 2 Membangkitkan populasi kemungkinan jawaban awal secara acak
 - 3 Mengevaluasi fungsi objektif
 - 4 Melakukan operasi terhadap kromosom menggunakan operator genetik (reproduksi, kawin silang, dan mutasi)
 - 5 Ulangi langkah 3 dan 4 sampai mencapai kriteria untuk menghentikan algoritma.

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

- Dasar Teori

—Algoritma *Hybrid Genetic*

– Cara Kerja Algoritma Genetik

Cara kerja algoritma genetik adalah sebagai berikut [?]:

1. Menentukan populasi kromosom kemungkinan jawaban awal.
2. Membangkitkan populasi kemungkinan jawaban awal secara acak.
3. Mengevaluasi fungsi objektif.
4. Melakukan operasi terhadap kromosom menggunakan operator genetika (reproduksi, kawin silang, dan mutasi).
5. Ulangi langkah 3 dan 4 sampai mencapai kriteria untuk menghentikan algoritma.

- 1 Cara kerja algoritma genetik adalah sebagai berikut:
- 2 Menentukan populasi kromosom kemungkinan jawaban awal
- 3 Membangkitkan populasi kemungkinan jawaban awal secara acak
- 4 Mengevaluasi fungsi objektif
- 5 Melakukan operasi terhadap kromosom menggunakan operator genetik (reproduksi, kawin silang, dan mutasi)
- 6 Ulangi langkah 3 dan 4 sampai mencapai kriteria untuk menghentikan algoritma.

Langkah-Langkah Utama dalam Penggunaan Algoritma Genetik

- Langkah-langkah utama dalam penggunaan algoritma genetik adalah:
- Membangkitkan populasi kemungkinan jawaban
- Mencari fungsi objektif dan fungsi kelayakan
- Penggunaan operator genetik

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

- Dasar Teori

—Algoritma *Hybrid Genetic*

—Langkah-Langkah Utama dalam Penggunaan Algoritma Genetik

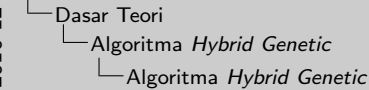
Langkah-langkah utama dalam penggunaan algoritma genetik adalah membangkitkan populasi kemungkinan jawaban, mencari fungsi objektif dan fungsi kelayakan, dan penggunaan operator genetik.

Algoritma *Hybrid Genetic*

- Pencarian *rule based* dimulai dengan mengasumsikan semua nilai sel yang tidak diketahui dengan semua kemungkinan nilai untuk mengisi sel tersebut tanpa melanggar batasan, dengan $P(C_{b,k}) = 1, 2, \dots, n$
- Setelah nilai dari satu sel sudah ditentukan, kemungkinan nilai untuk beberapa sel tertentu diperbaharui
- Misalnya, penggunaan aturan *naked single* yang dinyatakan dalam persamaan 1 di slide berikutnya, akan mengakibatkan semua kemungkinan nilai untuk semua sel lain dalam baris yang sama dan dalam kolom yang sama harus diperbaharui, seperti dinyatakan dalam persamaan 2 dan 3 di slide berikutnya
- Aturan *naked pair*, salah satu dari aturan jenis *naked subset*, dinyatakan dalam persamaan 4 untuk baris dan persamaan 5 untuk kolom

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku



Pencarian *rule based* dimulai dengan mengasumsikan semua nilai sel yang tidak diketahui dengan semua kemungkinan nilai untuk mengisi sel tersebut tanpa melanggar batasan, dengan $P(C_{b,k}) = 1, 2, \dots, n$. Setelah nilai dari satu sel sudah ditentukan, kemungkinan nilai untuk beberapa sel tertentu diperbaharui. Misalnya, penggunaan aturan *naked single* yang dinyatakan dalam persamaan 1 di bawah ini, akan mengakibatkan semua kemungkinan nilai untuk semua sel lain dalam baris yang sama dan dalam kolom yang sama harus diperbaharui, seperti dinyatakan dalam persamaan 2 dan 3 di bawah ini. Aturan *naked pair*, salah satu dari aturan jenis *naked subset*, dinyatakan dalam persamaan 4 untuk baris dan persamaan 5 untuk kolom. [?]

- Pencarian *rule based* ditulud dengan mengasumsikan semua nilai *set* yang tidak diketahui dengan semua kemungkinan nilai untuk mengimisi *set* tersebut tanpa memerlukan batasan, dengan $P(S_{i,j}) = 1, 2, \dots, n$
- Setelah nilai dari satu *set* sudah ditentukan, kemungkinan nilai untuk *set* lainnya sudah tertentu diperbarui
- Misalnya, penggunaan aturan *naked single* yang dinyatakan dalam persamaan 1 di *side berikutnya*, akan mengakibatkan semua kemungkinan nilai untuk semua *set* lain dalam baris yang sama dan dalam kolom yang sama harus diperbarui, seperti dinyatakan dalam persamaan 2 dan 3 di *side berikutnya*
- Aturan *naked pair*, salah satu dari aturan jenis *naked subset*, dinyatakan dalam persamaan 4 untuk baris dan persamaan 5 untuk kolom

Algoritma *Hybrid Genetic*

- 1 $|P(C_{b,k})| = 1 \wedge x \in P(C_{b,k}) \rightarrow V(C_{b,k}) = x$, artinya jika sebuah *cage* berukuran 1 sel, dan x adalah nilai tujuan dari *cage* tersebut, maka nilai dari sel tersebut adalah x
- 2 $(V(C_{b,k}) = x) \wedge (\forall a \in \{1, 2, \dots, n\}) \rightarrow P(C_{a,k}) = P(C_{a,k}) - \{x\}$, artinya jika nilai suatu sel pada baris b dan kolom k adalah x , maka x dihapus dari kemungkinan angka-angka yang bisa digunakan untuk mengisi sel-sel lain pada baris b
- 3 $(V(C_{b,k}) = x) \wedge (\forall q \in \{1, 2, \dots, n\}) \rightarrow P(C_{b,q}) = P(C_{b,q}) - \{x\}$ artinya jika nilai suatu sel pada baris b dan kolom k adalah x , maka x dihapus dari kemungkinan angka-angka yang bisa digunakan untuk mengisi sel-sel lain pada kolom k

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan *Calcludoku*

Dasar Teori

Algoritma *Hybrid Genetic*

Algoritma *Hybrid Genetic*

1. $|P(C_{b,k})| = 1 \wedge x \in P(C_{b,k}) \rightarrow V(C_{b,k}) = x$, artinya jika sebuah *cage* berukuran 1 sel, dan x adalah nilai tujuan dari *cage* tersebut, maka nilai dari sel tersebut adalah x .
2. $(V(C_{b,k}) = x) \wedge (\forall a \in \{1, 2, \dots, n\}) \rightarrow P(C_{a,k}) = P(C_{a,k}) - \{x\}$, artinya jika nilai suatu sel pada baris b dan kolom k adalah x , maka x dihapus dari kemungkinan angka-angka yang bisa digunakan untuk mengisi sel-sel lain pada baris b .
3. $(V(C_{b,k}) = x) \wedge (\forall q \in \{1, 2, \dots, n\}) \rightarrow P(C_{b,q}) = P(C_{b,q}) - \{x\}$ artinya jika nilai suatu sel pada baris b dan kolom k adalah x , maka x dihapus dari kemungkinan angka-angka yang bisa digunakan untuk mengisi sel-sel lain pada kolom k .

Algoritma *Hybrid Genetic*

- 1 $|P(C_{b,k})| = 1 \wedge x \in P(C_{b,k}) \rightarrow V(C_{b,k}) = x$, artinya jika sebuah *cage* berukuran 1 sel, dan x adalah nilai tujuan dari *cage* tersebut, maka nilai dari sel tersebut adalah x
- 2 $(V(C_{b,k}) = x) \wedge (\forall a \in \{1, 2, \dots, n\}) \rightarrow P(C_{a,k}) = P(C_{a,k}) - \{x\}$, artinya jika nilai suatu sel pada baris b dan kolom k adalah x , maka x dihapus dari kemungkinan angka-angka yang bisa digunakan untuk mengisi sel-sel lain pada baris b
- 3 $(V(C_{b,k}) = x) \wedge (\forall q \in \{1, 2, \dots, n\}) \rightarrow P(C_{b,q}) = P(C_{b,q}) - \{x\}$ artinya jika nilai suatu sel pada baris b dan kolom k adalah x , maka x dihapus dari kemungkinan angka-angka yang bisa digunakan untuk mengisi sel-sel lain pada kolom k

Algoritma *Hybrid Genetic*

4

$|P(C_{b,k1})| = |P(C_{b,k2})| = 2 \wedge P(C_{b,k1}) = P(C_{b,k2}) \rightarrow P(C_{b,q}) = P(C_{b,q}) - P(C_{b,k1})$, artinya jika ada dua sel dalam satu baris yang hanya bisa diisi oleh dua kemungkinan angka, maka kedua angka tersebut dihapus dari kemungkinan angka-angka yang bisa digunakan untuk mengisi sel-sel lain pada baris tersebut

5

$|P(C_{b1,k})| = |P(C_{b2,k})| = 2 \wedge P(C_{b1,k}) = P(C_{b2,k}) \rightarrow$
 $P(C_{p,k}) = P(C_{p,k}) - P(C_{b1,k})$, artinya jika ada dua sel
 dalam satu kolom yang hanya bisa diisi oleh dua
 kemungkinan angka, maka kedua angka tersebut dihapus
 dari kemungkinan angka-angka yang bisa digunakan
 untuk mengisi sel-sel lain pada kolom tersebut

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

- Dasar Teori

– Algoritma *Hybrid Genetic*

— Algoritma *Hybrid Genetic*

4. $|P(C_{b,k1})| = |P(C_{b,k2})| = 2 \wedge P(C_{b,k1}) = P(C_{b,k2}) \rightarrow P(C_{b,q}) = P(C_{b,q}) - P(C_{b,k1})$, artinya jika ada dua sel dalam satu baris yang hanya bisa diisi oleh dua kemungkinan angka, maka kedua angka tersebut dihapus dari kemungkinan angka-angka yang bisa digunakan untuk mengisi sel-sel lain pada baris tersebut.

5. $|P(C_{b1,k})| = |P(C_{b2,k})| = 2 \wedge P(C_{b1,k}) = P(C_{b2,k}) \rightarrow P(C_{p,k}) = P(C_{p,k}) - P(C_{b1,k})$, artinya jika ada dua sel dalam satu kolom yang hanya bisa diisi oleh dua kemungkinan angka, maka kedua angka tersebut dihapus dari kemungkinan angka-angka yang bisa digunakan untuk mengisi sel-sel lain pada kolom tersebut.

2. $|P(C_{2A2})| = |P(C_{2A2}) - 2 \cdot P(C_{1A1}) \cdot P(C_{2A2})| = P(C_{2A2}) - P(C_{1A1}) \cdot P(C_{2A2})$, artinya jika ada dua sel dalam satu baris yang hanya bisa diisi oleh dua kemungkinan angka, maka kedua angka tersebut dihapus dari kemungkinan angka-angka yang bisa digunakan untuk mengisi sel-sel lain pada baris tersebut

3. $|P(C_{3A1})| = |P(C_{3A2}) - 2 \cdot P(C_{1A1}) \cdot P(C_{3A2})| = P(C_{3A1}) - P(C_{1A1}) \cdot P(C_{3A1})$, artinya jika ada dua sel dalam satu kolom yang hanya bisa diisi oleh dua kemungkinan angka, maka kedua angka tersebut dihapus dari kemungkinan angka-angka yang bisa digunakan untuk mengisi sel-sel lain pada kolom tersebut

Algoritma *Hybrid Genetic*

- Algoritma genetik digunakan saat teka-teki masih tidak bisa diselesaikan setelah mengerjakan semua aturan logika secara berulang-ulang
- Algoritma ini dimulai dengan meng-*encode* kromosom

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

- Dasar Teori

—Algoritma *Hybrid Genetic*

—Algoritma *Hybrid Genetic*

Algoritma genetik digunakan saat teka-teki masih tidak bisa diselesaikan setelah mengerjakan semua aturan logika secara berulang-ulang. Algoritma ini dimulai dengan meng-*encode* kromosom.

- Algoritma genetika digunakan saat teka-teki masih tidak bisa diselesaikan setelah mengerjakan semua aturan logika secara berulang-ulang
- Algoritma ini dimulai dengan meng-encode kromosom

Pemodelan Kromosom

- Satu kromosom terdiri dari k segmen, dengan $m \leq n$
- Satu segmen berisikan sekumpulan gen yang belum diselesaikan yang berada di dalam segmen tersebut
- Sebuah segmen merepresentasikan sebuah baris
- Dalam sebuah kromosom, segmen diurutkan dari baris yang paling atas ke baris yang paling bawah
- Setiap segmen merepresentasikan sebuah baris yang belum terselesaikan

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

- Dasar Teori

—Algoritma *Hybrid Genetic*

–Pemodelan Kromosom

Satu kromosom terdiri dari k segmen, dengan $m \leq n$. Satu segmen berisikan sekumpulan gen yang belum diselesaikan yang berada di dalam segmen tersebut. Sebuah segmen merepresentasikan sebuah baris atau kolom. Dalam sebuah kromosom, segmen diurutkan dari baris yang paling atas ke baris yang paling bawah atau dari kolom yang paling kiri ke kolom yang paling kanan. Setiap segmen merepresentasikan sebuah baris yang belum terselesaikan.

- Satu kromosom terdiri dari k segmen, dengan $m \leq n$
- Satu segmen berisikan sekumpulan gen yang belum diselesaikan yang berada di dalam segmen tersebut
- Sebuah segmen merepresentasikan sebuah baris
- Dalam sebuah kromosom, segmen diurutkan dari baris yang paling atas ke baris yang paling bawah
- Setiap segmen merepresentasikan sebuah baris yang belum terselesaikan

Pemodelan Kromosom

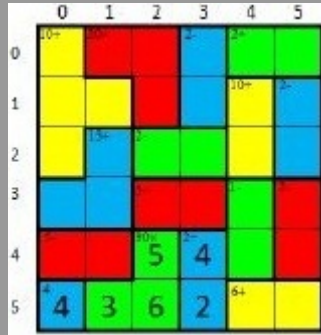


Figure 9: Contoh permainan teka-teki Calcudoku dengan ukuran *grid* 6 x 6

Contoh salah satu kromosom dari permainan teka-teki Calcudoku pada Gambar 9 adalah:

34 35 | 28 29 24 25 | 0 4 5 1 2 3 | 11 6 9 7 8 10 |
12 14 15 17 16 13 | 20 18 19 23 21 22

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

└ Dasar Teori

└ Algoritma *Hybrid Genetic*

└ Pemodelan Kromosom

Contoh, salah satu kromosom dari permainan teka-teki Calcudoku pada Gambar 9 adalah

34 35 | 28 29 24 25 | 0 4 5 1 2 3 | 11 6 9 7 8 10 | 12 14 15 17 16 13 | 20 18 19 23 21 22.

Pemodelan Kromosom



Figure 9: Contoh permainan teka-teki Calcudoku dengan ukuran *grid* 6 x 6

Contoh salah satu kromosom dari permainan teka-teki Calcudoku pada Gambar 9 adalah:

34 35 | 28 29 24 25 | 0 4 5 1 2 3 | 11 6 9 7 8 10 |
12 14 15 17 16 13 | 20 18 19 23 21 22

Fungsi Objektif

- Fungsi objektif, yang direpresentasikan dengan x_j , akan dihitung setelah pembangkitan nilai dari gen pada kromosom sudah dilakukan
- Nilai untuk gen ke- j pada sebuah kromosom direpresentasikan dengan w_j
- x_j akan bernilai 0 jika belum diselesaikan ($w_j = 0$), dan bernilai 1 jika sudah diselesaikan ($w_j \neq 0$)

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

- Dasar Teori

—Algoritma *Hybrid Genetic*

— Fungsi Objektif

Menurut Johanna, Lukas, dan Saputra, fungsi objektif, yang direpresentasikan dengan x_j , akan dihitung setelah pembangkitan nilai dari gen pada kromosom sudah dilakukan. Nilai untuk gen ke- j pada sebuah kromosom direpresentasikan dengan w_j . x_j akan bernilai 0 jika belum diselesaikan ($w_j = 0$), dan bernilai 1 jika sudah diselesaikan ($w_j \neq 0$).

- Fungsi objektif, yang direpresentasikan dengan x_j , akan dihitung setelah pembangkitan nilai dari gen pada kromosom sudah dilakukan
- Nilai untuk gen ke- j pada sebuah kromosom direpresentasikan dengan w_j
- x_j akan bernilai 0 jika belum diselesaikan ($w_j = 0$), dan bernilai 1 jika sudah diselesaikan ($w_j \neq 0$)

Fungsi Kelayakan

- Untuk kromosom dengan jumlah gen k , fungsi kelayakan, yaitu hasil penjumlahan dari hasil fungsi objektif untuk setiap gen dibagi dengan jumlah gen, dinyatakan dalam persamaan di bawah ini:

$$x_j = \begin{cases} 0, & w_j = 0 \\ 1, & w_j \neq 0 \end{cases}$$

$$fitness = \frac{\sum_{j=0}^k x_j}{k}$$

- Jadi, solusi dari teka-teki ini adalah mencari kromosom yang nilai kelayakannya 1

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

└ Dasar Teori

└ Algoritma *Hybrid Genetic*

└ Fungsi Kelayakan

Fungsi Kelayakan

- Untuk kromosom dengan jumlah gen k , fungsi kelayakan, yaitu hasil penjumlahan dari hasil fungsi objektif untuk setiap gen dibagi dengan jumlah gen, dinyatakan dalam persamaan di bawah ini:

$$x_j = \begin{cases} 0, & w_j = 0 \\ 1, & w_j \neq 0 \end{cases}$$

$$fitness = \frac{\sum_{j=0}^k x_j}{k}$$

- Jadi, solusi dari teka-teki ini adalah mencari kromosom yang nilai kelayakannya 1

Fungsi kelayakan, yaitu hasil penjumlahan dari hasil fungsi objektif untuk setiap gen dibagi dengan jumlah gen, dinyatakan dalam persamaan di bawah ini [?]:

$$x_j = \begin{cases} 0, & w_j = 0 \\ 1, & w_j \neq 0 \end{cases}$$

$$fitness = \frac{\sum_{j=0}^k x_j}{k}$$

Jadi, solusi dari teka-teki ini adalah mencari kromosom yang nilai kelayakannya 1.

Kawin Silang

Parent 1 :	34	35	28	29	24	25	0	4	5	1	2	3	11	6	9	7	8	10	12	14	15	17	16	13	20	18	19	23	21	22
Parent 2 :	34	35	29	25	24	28	3	5	4	2	0	1	10	11	7	8	6	9	17	15	14	16	13	12	22	20	19	23	18	21
Child 1 :	34	35	29	25	24	28	0	4	5	1	2	3	11	6	9	7	8	10	17	15	14	16	13	12	20	18	19	23	21	22
Child 2 :	34	35	28	29	24	25	3	5	4	2	0	1	10	11	7	8	6	9	12	14	15	17	16	13	22	20	19	23	18	21

Figure 10: Contoh proses kawin silang antara dua kromosom
[?]

- Dua kromosom, yaitu kromosom orang tua, disilangkan untuk membuat dua kromosom yang baru, yaitu kromosom anak, dengan metodologi kawin silang *N-segments*
- Gambar 10 menggambarkan contoh proses kawin silang antara dua kromosom.

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

└ Dasar Teori

└ Algoritma *Hybrid Genetic*

└ Kawin Silang

Kawin Silang



Figure 10: Contoh proses kawin silang antara dua kromosom
[?]

- Dua kromosom, yaitu kromosom orang tua, disilangkan untuk membuat dua kromosom yang baru, yaitu kromosom anak, dengan metodologi kawin silang *N-segments*
- Gambar 10 menggambarkan contoh proses kawin silang antara dua kromosom.

Dalam proses reproduksi kawin silang, dua kromosom, yaitu kromosom orang tua, disilangkan untuk membuat dua kromosom yang baru, yaitu kromosom anak, dengan metodologi kawin silang *N-segments*. Gambar 10 menggambarkan contoh proses kawin silang antara dua kromosom.

Mutasi

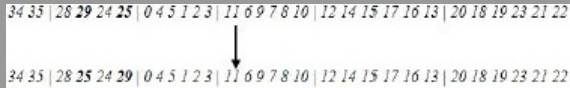


Figure 11: Contoh proses mutasi [?]

- Digunakan untuk mendapatkan kemungkinan kromosom yang lain
- Dilakukan di antara gen yang berada dalam segmen yang sama
- Gambar 11 adalah contoh proses mutasi antara dua gen dalam segmen yang sama

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

└ Dasar Teori

└ Algoritma *Hybrid Genetic*

└ Mutasi

Pertukaran mutasi digunakan untuk mendapatkan kemungkinan kromosom yang lain. Mutasi dilakukan di antara gen yang berada dalam segmen yang sama. Gambar 11 adalah contoh proses mutasi antara dua gen dalam segmen yang sama.

Mutasi



Figure 11: Contoh proses mutasi [?]

- Digunakan untuk mendapatkan kemungkinan kromosom yang lain
- Dilakukan di antara gen yang berada dalam segmen yang sama
- Gambar 11 adalah contoh proses mutasi antara dua gen dalam segmen yang sama

Cara Kerja Algoritma *Hybrid Genetic*

- Cara kerja algoritma *hybrid genetic* menurut Johanna, Lukas, dan Saputra adalah sebagai berikut:
- Masukkan teka-teki yang akan diselesaikan sebagai input.
- Program akan merepresentasikan input yang dimasukkan dalam format teka-teki.
- Program akan mencoba menyelesaikan teka-teki tersebut dengan menggunakan algoritma *rule based* terlebih dahulu.
- Jika program berhasil menyelesaikan teka-teki tersebut dengan menggunakan algoritma *rule based*, maka algoritma selesai.
- Jika program gagal dengan menggunakan algoritma *rule based*, maka program akan mencoba menyelesaikan teka-teki tersebut dengan menggunakan algoritma genetik.
- Jika program berhasil menyelesaikan teka-teki tersebut dengan menggunakan algoritma genetik, maka algoritma selesai.
- Jika program gagal dalam menyelesaikan teka-teki tersebut setelah menggunakan algoritma genetik, artinya algoritma gagal dalam menyelesaikan teka-teki tersebut.

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

└ Dasar Teori

└ Algoritma *Hybrid Genetic*

└ Cara Kerja Algoritma *Hybrid Genetic*

Cara kerja algoritma *hybrid genetic* adalah sebagai berikut [?]:

- Masukkan teka-teki yang akan diselesaikan sebagai input.
- Program akan merepresentasikan input yang dimasukkan dalam format teka-teki.
- Program akan mencoba menyelesaikan teka-teki tersebut dengan menggunakan algoritma *rule based* terlebih dahulu.
- Jika program berhasil menyelesaikan teka-teki tersebut dengan menggunakan algoritma *rule based*, maka algoritma selesai.
- Jika program gagal dengan menggunakan algoritma *rule based*, maka program akan mencoba menyelesaikan teka-teki tersebut dengan menggunakan algoritma genetik.
- Jika program berhasil menyelesaikan teka-teki tersebut dengan menggunakan algoritma genetik, maka algoritma selesai.
- Jika program gagal dalam menyelesaikan teka-teki tersebut setelah menggunakan algoritma genetik, artinya algoritma gagal dalam menyelesaikan teka-teki tersebut.

Cara Kerja Algoritma *Hybrid Genetic*

- Cara kerja algoritma *hybrid genetic* menurut Johanna, Lukas, dan Saputra adalah sebagai berikut:
- Masukkan teka-teki yang akan diselesaikan sebagai input.
- Program akan merepresentasikan input yang dimasukkan dalam format teka-teki.
- Program akan mencoba menyelesaikan teka-teki tersebut dengan menggunakan algoritma *rule based* terlebih dahulu.
- Jika program berhasil menyelesaikan teka-teki tersebut dengan menggunakan algoritma *rule based*, maka algoritma selesai.
- Jika program gagal dengan menggunakan algoritma *rule based*, maka program akan mencoba menyelesaikan teka-teki tersebut dengan menggunakan algoritma genetik.
- Jika program berhasil menyelesaikan teka-teki tersebut dengan menggunakan algoritma genetik, maka algoritma selesai.
- Jika program gagal dalam menyelesaikan teka-teki tersebut setelah menggunakan algoritma genetik, artinya algoritma gagal dalam menyelesaikan teka-teki tersebut.

Alur Cara Kerja Algoritma *Hybrid Genetic*

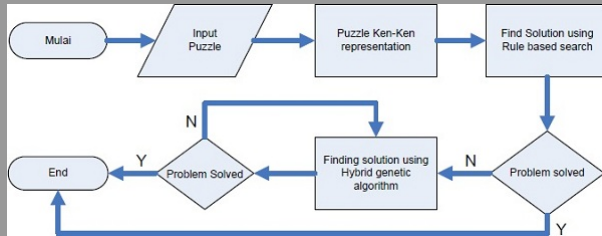


Figure 12: Alur penyelesaian permainan teka-teki Calcudoku dengan menggunakan algoritma *hybrid genetic*

- Alur (*flow chart*) penyelesaian permainan teka-teki Calcudoku dengan menggunakan algoritma *hybrid genetic* dapat dilihat di Gambar 12.

2016-12-04

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

└ Dasar Teori

└ Algoritma *Hybrid Genetic*

└ Alur Cara Kerja Algoritma *Hybrid Genetic*

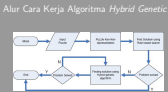


Figure 12: Alur penyelesaian permainan teka-teki Calcudoku dengan menggunakan algoritma *hybrid genetic*

- Alur (*flow chart*) penyelesaian permainan teka-teki Calcudoku dengan menggunakan algoritma *hybrid genetic* dapat dilihat di Gambar 12.

Alur (*flow chart*) penyelesaian permainan teka-teki Calcudoku dengan menggunakan algoritma *hybrid genetic* dapat dilihat di Gambar 12.