

PENDALAMAN MATERI - KELOMPOK 5

KECERDASAN KOMPUTASIONAL

Dosen Pengampu : Dr. Wiharto, S.T., M.Kom.



Disusun Oleh :

M0521033 – Irzan Rafi Imtinan

M0521048 – Muhammad Azzam Hilmy

M0521050 – Muhammad Fauzan Hidayat

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI DAN SAINS DATA

UNIVERSITAS SEBELAS MARET

2024

PENDALAMAN MATERI

1. Pada jenis permasalahan apa algoritma heuristik seharusnya diterapkan?

Jawab:

Algoritma heuristik seharusnya diterapkan pada jenis permasalahan yang kompleks dan memiliki ruang solusi yang sangat besar, di mana pencarian solusi optimum melalui metode eksak (seperti algoritma deterministik) akan memakan waktu yang sangat lama atau terlalu sulit secara matematis. Beberapa jenis permasalahan yang cocok untuk algoritma heuristik adalah:

1. Masalah Optimasi Kombinatorik: Seperti masalah penjadwalan, rute terpendek, atau masalah transportasi, di mana ada banyak kombinasi kemungkinan solusi dan pencarian solusi optimal dengan metode eksak bisa memakan waktu yang tidak realistis.
2. Masalah dengan Batasan Kompleks: Masalah yang memiliki banyak batasan (seperti dalam komposisi makanan ternak atau distribusi barang) yang sulit dianalisis secara eksak dan membutuhkan solusi cepat yang mendekati optimal.
3. Masalah Dimensi Besar: Masalah yang melibatkan data dalam skala besar atau banyak entitas, seperti pengaturan jadwal kuliah atau distribusi komoditas dalam jaringan besar, di mana pencarian solusi eksak membutuhkan perhitungan yang panjang.
4. Masalah yang Tidak Terdefinisi dengan Jelas atau Mengandung Ketidakpastian: Dalam situasi di mana tidak semua variabel dapat diketahui atau diukur dengan pasti, atau masalah yang bersifat dinamis, metode heuristik dapat membantu mencari solusi yang memadai meskipun tidak selalu optimal.

Algoritma heuristik tidak selalu memberikan solusi optimum, tetapi dirancang untuk menemukan solusi yang cukup baik dalam waktu yang lebih singkat.

2. Jelaskan apa yang dimaksud dengan gen, individu, populasi, generasi dalam algoritma evolusi!

Jawab:

Dalam algoritma evolusi (Evolutionary Algorithms, EAs), konsep gen, individu, populasi, dan generasi digunakan untuk meniru proses evolusi biologi. Berikut adalah penjelasan dan informasi tentang masing – masing konsep tersebut:

1. Gen: Dalam algoritma evolusi, gen merepresentasikan unit dasar dari solusi masalah. Gen adalah bagian terkecil dari individu yang menggambarkan satu aspek tertentu

dari solusi. Misalnya, dalam masalah optimasi rute perjalanan, gen bisa merepresentasikan satu kota dalam urutan perjalanan. Setiap gen menyimpan nilai yang berkontribusi terhadap keseluruhan solusi.

2. Individu: Individu adalah kombinasi dari sejumlah gen yang membentuk satu solusi penuh terhadap masalah yang sedang diselesaikan. Dalam konteks algoritma evolusi, setiap individu merepresentasikan sebuah kandidat solusi. Misalnya, dalam masalah Traveling Salesman Problem (TSP), satu individu bisa menggambarkan satu rute perjalanan lengkap dari satu kota ke kota lain.
3. Populasi: Populasi adalah kumpulan dari individu-individu yang dipertimbangkan dalam satu iterasi (generasi) algoritma. Pada setiap langkah, algoritma evolusi akan bekerja dengan populasi ini untuk mencari solusi yang lebih baik. Populasi bertindak sebagai ruang pencarian di mana algoritma akan mengevaluasi, mengkombinasikan, dan mengubah individu-individu untuk menghasilkan solusi yang lebih optimal.
4. Generasi: Generasi adalah siklus dalam proses evolusi, dimana populasi baru individu dihasilkan melalui proses seleksi, reproduksi (seperti crossover atau mutasi), dan evaluasi. Setelah setiap generasi, populasi akan dievaluasi ulang untuk menentukan individu terbaik yang akan menghasilkan keturunan di generasi berikutnya. Dari generasi ke generasi, kualitas solusi yang dihasilkan diharapkan meningkat karena individu-individu terbaik dari generasi sebelumnya akan lebih mungkin terpilih.

Dengan kata lain, proses algoritma evolusi secara berulang menciptakan populasi baru, menguji seberapa baik individu dalam menyelesaikan masalah (menggunakan fungsi fitness), dan menggunakan individu terbaik untuk menciptakan generasi berikutnya, hingga mencapai solusi yang mendekati optimal atau memenuhi kriteria yang ditetapkan.

3. Apa yang dimaksud dengan fungsi fitness?

Jawab:

Fungsi fitness dalam algoritma evolusi adalah sebuah fungsi yang digunakan untuk mengevaluasi seberapa baik suatu individu (solusi) menyelesaikan masalah yang sedang dioptimalkan. Fungsi ini mengukur "kualitas" atau "kecocokan" individu tersebut terhadap tujuan optimasi.

Fungsi fitness dalam algoritma evolusi digunakan untuk mengevaluasi setiap individu dalam populasi dengan memberikan nilai numerik yang menunjukkan seberapa baik individu tersebut dalam memenuhi tujuan optimasi. Semakin tinggi nilai fitness, semakin baik solusi yang diberikan oleh individu tersebut.

Fungsi fitness juga memainkan peran penting dalam proses seleksi alam, di mana individu dengan nilai fitness yang lebih tinggi memiliki peluang lebih besar untuk dipilih sebagai "parent" yang akan menghasilkan keturunan melalui mekanisme seperti crossover dan mutasi. Ini meniru prinsip seleksi alam, di mana individu terbaik memiliki peluang lebih

besar untuk mewariskan sifat-sifatnya kepada generasi berikutnya. Algoritma evolusi secara bertahap meningkatkan populasi melalui beberapa generasi berdasarkan nilai fitness individu, dengan tujuan akhir menemukan solusi terbaik atau mendekati optimal.

Sebagai contoh, dalam masalah pencarian rute terpendek, fungsi fitness dapat berupa jarak total yang ditempuh. Solusi dengan rute terpendek akan memiliki nilai fitness yang lebih baik dibandingkan solusi dengan rute yang lebih panjang. Jadi, fungsi fitness adalah alat pengukur utama yang digunakan oleh algoritma evolusi untuk mengevaluasi solusi dan membimbing proses evolusi menuju hasil yang lebih optimal.

4. Apa yang dimaksud dengan pernyataan bahwa algoritma evolusi bersifat stochastic?

Jawab:

Algoritma evolusi dikatakan bersifat stochastic karena melibatkan elemen acak dalam proses pencarian solusinya. Pada setiap langkah, ada keputusan yang diambil secara acak, seperti pemilihan individu yang akan menjadi parent, penerapan crossover, atau pengenalan mutasi pada individu. Penggunaan bilangan acak ini membuat setiap iterasi algoritma memiliki variasi dalam hasilnya, sehingga solusi akhir yang dihasilkan dapat berbeda setiap kali algoritma dijalankan, meskipun permasalahan yang dihadapi sama dan kondisi awalnya identik.

Sifat stochastic ini memberikan keunggulan penting dalam eksplorasi ruang solusi. Karena proses pencarian solusi tidak sepenuhnya deterministik, algoritma evolusi memiliki peluang lebih besar untuk menghindari jebakan local optima — yaitu solusi yang tampaknya optimal dalam ruang pencarian lokal tetapi bukan solusi global terbaik. Dengan adanya elemen acak, algoritma evolusi dapat mengeksplorasi berbagai kemungkinan solusi dan mencari alternatif yang lebih baik di luar area yang sudah dieksplorasi sebelumnya.

Meskipun ada elemen acak, algoritma evolusi tetap diarahkan oleh fungsi fitness. Setelah proses acak dilakukan, individu-individu terbaik tetap dipilih untuk reproduksi dan evolusi selanjutnya. Dengan demikian, elemen stochastic dalam algoritma evolusi menciptakan keseimbangan antara eksplorasi ruang solusi yang luas dan pencarian solusi yang mendekati optimal berdasarkan kriteria yang ditetapkan.

5. Jelaskan perbedaan antara Soft Constraints dan Hard Constraints, dan berikan contohnya!

Jawab:

Soft Constraints dan Hard Constraints adalah dua jenis batasan dalam masalah optimasi yang menentukan seberapa ketat aturan atau kondisi yang harus dipenuhi oleh solusi. Berikut adalah perbedaan utama antara keduanya beserta contohnya:

1. Hard Constraints:

Hard constraints adalah batasan yang wajib dipenuhi dalam sebuah solusi. Jika solusi melanggar hard constraints, maka solusi tersebut dianggap tidak valid atau tidak dapat diterima. Batasan ini bersifat absolut dan tidak bisa dilanggar.

Contoh: Dalam masalah penjadwalan kuliah, sebuah hard constraint bisa berupa aturan bahwa seorang dosen atau mahasiswa tidak boleh berada di dua kelas yang berbeda pada waktu yang sama. Pelanggaran terhadap batasan ini akan menyebabkan jadwal tersebut tidak valid.

2. Soft Constraints:

Soft constraints adalah batasan yang sebaiknya dipenuhi, tetapi pelanggarannya masih dapat diterima meskipun mungkin mengurangi kualitas solusi. Solusi yang melanggar soft constraints tidak dianggap tidak valid, tetapi dianggap kurang optimal atau kurang baik dibandingkan solusi yang memenuhi batasan tersebut.

Contoh: Dalam masalah yang sama, sebuah soft constraint bisa berupa preferensi bahwa jadwal kuliah sebaiknya tidak berlangsung terlalu pagi atau terlalu sore. Jika jadwal kuliah diatur pada jam tersebut, solusinya masih valid, tetapi dianggap kurang ideal bagi dosen atau mahasiswa.

Sehingga hal ini mengartikan, Hard constraints harus dipenuhi untuk menghasilkan solusi yang valid, sementara soft constraints adalah preferensi yang jika dilanggar, tetap menghasilkan solusi yang valid tetapi kurang optimal.

TUGAS KELOMPOK

1. Sebutkan semua proses utama dalam siklus algoritma genetika?

Jawab:

- a. Inisialisasi Populasi
 - Membuat populasi awal, Populasi awal berisi sejumlah individu (kromosom) yang merepresentasikan solusi potensial untuk masalah.
 - Representasi, Setiap individu biasanya diwakili oleh string bit (0 dan 1) atau struktur data lain, yang mewakili karakteristik solusi.
 - Inisialisasi Acak, Individu-individu dalam populasi awal biasanya diinisialisasi secara acak untuk memastikan keragaman dalam populasi.
- b. Evaluasi Kecocokan (Fitness)
 - Fungsi fitness adalah fungsi yang menentukan seberapa baik setiap individu dalam populasi menyelesaikan masalah.
 - Nilai Kecocokan fungsi fitness memberikan nilai numerik untuk setiap individu, di mana nilai yang lebih tinggi menunjukkan solusi yang lebih baik.
 - Tujuannya adalah untuk memaksimalkan nilai fitness dalam populasi.
- c. Seleksi:
 - Memilih Individu dengan nilai fitness yang lebih tinggi memiliki peluang lebih besar untuk dipilih untuk bereproduksi.
 - Menyeleksi dengan teknik seleksi, seperti seleksi roulette wheel, seleksi turnamen, dan seleksi peringkat.
- d. Crossover (Perkawinan):
 - Menggabungkan Gen yaitu dengan cara memilih individu yang terpilih direproduksi dengan cara menggabungkan gen-gen mereka untuk menciptakan keturunan baru.
 - Menyilangkan atau mengawinkan menggunakan Teknik Crossover seperti single-point crossover, two-point crossover, dan uniform crossover.
- e. Mutasi:
 - Mutasi adalah perubahan acak pada gen-gen individu yang dapat menghasilkan solusi baru.
 - Tujuan Mutasi untuk membantu menjaga keragaman dalam populasi dan mencegah algoritma terperangkap dalam minimum lokal.
- f. Iterasi:
 - Iterasi Proses yaitu dengan Langkah-langkah seleksi, crossover, dan mutasi diulang selama beberapa generasi.
 - Konvergensi algoritma biasanya akan terus melakukan iterasi sampai kondisi berhenti terpenuhi, seperti mencapai batas jumlah generasi atau mencapai nilai fitness yang diinginkan.

2. Jelaskan keunggulan algoritma genetika sebagai algoritma yang berbasis populasi!

Jawab:

Algoritma genetika (Genetic Algorithm, GA) memiliki beberapa keunggulan karena berbasis populasi dalam proses pencariannya. Berikut penjelasan utama mengenai keunggulan tersebut:

- **Kemampuan Eksplorasi Ruang Pencarian yang Luas**

Algoritma genetika melakukan pencarian solusi dari beberapa titik di ruang pencarian (search space) secara simultan, bukan hanya dari satu titik saja. Setiap individu dalam populasi merepresentasikan solusi yang berbeda. Dengan pendekatan ini, GAs dapat mengeksplorasi lebih banyak area dalam ruang pencarian, memungkinkan algoritma untuk mencari solusi secara lebih efektif di seluruh ruang pencarian. Hal ini sangat berguna dalam masalah yang kompleks atau memiliki banyak dimensi, karena GAs tidak mudah terjebak di optimum lokal seperti metode pencarian deterministik.

- **Diversitas Populasi**

Dalam GAs, populasi terdiri dari berbagai solusi yang bervariasi. Variasi ini dipertahankan melalui mekanisme reproduksi seperti crossover dan mutasi, yang menjaga keragaman populasi dan memungkinkan GAs untuk menghasilkan solusi yang lebih baik. Selain itu, GAs dapat membagi individu ke dalam sub-populasi, yang bisa diproses secara paralel atau di satu komputer, untuk memperluas eksplorasi ruang pencarian dan meningkatkan kualitas solusi. Penggunaan sub-populasi juga menjaga agar populasi tidak terlalu homogen, sehingga meminimalisir risiko terjebak di solusi lokal yang kurang optimal.

- **Solusi Multi-Objektif**

GAs mampu menghasilkan himpunan solusi optimal dalam satu siklus iterasi. Karena GAs bekerja dengan populasi yang beragam, algoritma ini dapat memberikan berbagai solusi yang hampir optimal atau optimal secara bersamaan. Ini sangat berguna dalam masalah dengan banyak tujuan (multi-objective optimization), di mana terdapat lebih dari satu kriteria yang harus dioptimalkan. Dengan demikian, pengguna dapat memilih solusi yang paling sesuai berdasarkan kebutuhan spesifik dari berbagai solusi yang telah dihasilkan.

Keunggulan-keunggulan ini menjadikan algoritma genetika sebagai salah satu pendekatan yang sangat kuat untuk masalah optimasi yang luas, kompleks, dan sulit dipecahkan dengan metode tradisional.

3. Jelaskan apa yang dimaksud dengan algoritma genetika bersifat ergodic?

Jawab:

Dalam konteks algoritma genetika, ergodic berarti bahwa algoritma ini memiliki kemampuan untuk mengeksplorasi seluruh ruang pencarian (search space) secara menyeluruh seiring berjalannya waktu. Dengan kata lain, algoritma genetika bersifat ergodic karena, secara teoritis, algoritma ini dapat menjelajahi setiap kemungkinan solusi dalam ruang pencarian jika diberikan waktu yang cukup panjang.

Sifat ergodic pada algoritma genetika merujuk pada kemampuannya untuk mengeksplorasi seluruh ruang pencarian secara menyeluruh seiring berjalannya waktu. Dengan menggunakan mekanisme seperti crossover, mutasi, dan seleksi acak, algoritma ini tidak terbatas pada area pencarian tertentu; setiap generasi baru membawa variasi yang memungkinkan populasi solusi menjelajahi bagian lain dari ruang pencarian, termasuk solusi yang belum dieksplorasi sebelumnya.

Selain itu, sifat ergodic memberikan algoritma genetika kemampuan untuk keluar dari solusi optimum lokal, sehingga meningkatkan peluang untuk menemukan solusi yang lebih baik. Meskipun tidak menjamin penemuan solusi optimal, jika diberikan cukup waktu dan generasi, algoritma ini akan mendekati solusi optimal karena aksesnya ke seluruh ruang pencarian, meskipun proses ini dapat memakan waktu yang lama dan sering kali dihentikan setelah menemukan solusi yang cukup baik.

Dengan demikian, sifat ergodic pada algoritma genetika menunjukkan bahwa algoritma ini mampu menjelajahi seluruh ruang solusi secara menyeluruh dan bertahap, sehingga memberikan peluang yang lebih baik untuk menemukan solusi optimal atau mendekati optimal.

4. Iterasi GAs diulang terus sampai kondisi berhenti tercapai. Jelaskan beberapa kriteria untuk hal ini?

Jawab:

Beberapa kriteria bisa dipakai untuk hal ini sebagai berikut:

- Iterasi berhenti sampai generasi n . Nilai n ditentukan sebelumnya berdasarkan beberapa eksperimen pendahuluan. Semakin tinggi ukuran dan kompleksitas masalah maka nilai n semakin besar. Nilai n ditentukan sedemikian rupa sehingga konvergensi populasi tercapai dan akan sulit didapatkan solusi yang lebih baik setelah n iterasi.
- Iterasi berhenti setelah n generasi berurutan tidak dijumpai solusi yang lebih baik. Kondisi ini menunjukkan bahwa GAs sulit mendapatkan solusi yang lebih baik dan penambahan iterasi hanya membuang waktu.
- Iterasi berhenti setelah t satuan waktu tercapai. Ini biasa digunakan jika diinginkan untuk membandingkan performa dari beberapa algoritma.

Dalam implementasi praktis, kombinasi kondisi (1) dan (2) bisa dipakai.

5. Misalkan P1 dan P2 adalah parent untuk proses crossover. Tentukan offspring yang terbentuk jika dilakukan one-cut-point crossover pada titik ke-5.

P1	[0 0 1 1 0 0 1]
P2	[1 0 0 1 1 1 1]

Jawab:

Untuk melakukan one-cut-point crossover pada titik ke-5 antara parent P1 dan P2, kita perlu membagi kedua parent di titik tersebut.

1. Titik potong adalah ke-5.

2. Pisahkan kedua parent:

P1: [0 0 1 1 0 | 0 1]

P2: [1 0 0 1 1 | 1 1]

3. Gabungkan bagian sebelum dan setelah titik potong:

- Bagian sebelum titik potong (1-4): [0 0 1 1] dari P1 dan [1 0 0 1] dari P2.

- Bagian setelah titik potong (5-7): [0 1] dari P1 dan [1 1] dari P2.

4. Bentuk offspring:

- Offspring 1: Gabungkan bagian dari P1 sebelum titik potong dan bagian dari P2 setelah titik potong:

O1: [0 0 1 1 | 1 1] → O1 = [0 0 1 1 1 1]

- Offspring 2: Gabungkan bagian dari P2 sebelum titik potong dan bagian dari P1 setelah titik potong:

O2: [1 0 0 1 | 0 1] → O2 = [1 0 0 1 0 1]

Jadi, offspring yang terbentuk adalah:

O1 = [0 0 1 1 1 1]

O2 = [1 0 0 1 0 1]

6. Misalkan P adalah parent untuk proses mutasi. Tentukan offspring yang terbentuk jika dilakukan mutasi pada titik ke-5.

$$P \quad [0011001]$$

Jawab:

Jika dilakukan mutasi pada titik ke-5 dari parent $P = [0011001]$ maka pada proses ini, nilai pada posisi ke-5 (indeks 4) diubah dari 0 menjadi 1. Ini didasarkan pada asumsi alfabet gen adalah biner (0 atau 1) dan jenis mutasi pembalikan gen. Sehingga offspring yang terbentuk adalah $[0011101]$.

7. Untuk masalah maksimasi (mencari nilai maksimum) dari sebuah fungsi sebagai berikut:

$$\max, y = f(x) = -(x^2)/2 + 4x + 40, \quad 0 \leq x \leq 15$$

Lengkapi tabel berikut:

	chromosome	x	$y=f(x)$	fitness
P_1	[0001]			
P_2	[1100]			
P_3	[1011]			
P_4	[1111]			

Jawab:

Pertama, Representasi Kromosom

Berdasarkan gambar, kromosom direpresentasikan dalam bentuk biner 4-bit.yaitu sebanyak $2^4 = 16$ dengan nilai batasan yaitu, $0 \leq x \leq 15$.

Kedua, Inisialisasi Populasi

Populasi awal sudah diberikan dalam gambar dengan 4 kromosom:

P_1 : [0 0 0 1] P_2 : [1 1 0 0] P_3 : [1 0 1 1] P_4 : [1 1 1 1]

Ketiga, Dekode Kromosom(Biner ke desimal)

Mengubah representasi biner menjadi nilai desimal:

P1: 0001 = 1 P2: 1100 = 12 P3: 1011 = 11 P4: 1111 = 15

Keempat, Evaluasi Fitness:

menghitung nilai fitness menggunakan fungsi $y = f(x) = -(x^2)/2 + 4x + 40$:

$$P1: f(1) = -(1^2)/2 + 4(1) + 40 = 43.5$$

$$P2: f(12) = -(12^2)/2 + 4(12) + 40 = -2$$

$$P3: f(11) = -(11^2)/2 + 4(11) + 40 = 7.5$$

$$P4: f(15) = -(15^2)/2 + 4(15) + 40 = -12.5$$

Kelima, Seleksi:

Individu dengan nilai fitness tertinggi memiliki peluang lebih besar untuk terpilih. Dalam kasus ini, *P1 memiliki fitness tertinggi.*

Keenam, Crossover:

Pilih dua induk, (P1,P3) tentukan titik crossover

P1: 00|01 , P3: 10|11

Hasil crossover: *C1: 0011 C2: 1001*

Ketujuh, Mutasi:

Mutasi bit terakhir *C1: 0011 -> 0010*

Kedelapan, Bentuk Populasi Baru:

Gabungkan hasil crossover dan mutasi dengan beberapa individu terbaik dari populasi sebelumnya untuk membentuk populasi baru.

P1 [0 0 0 1] (individu elit dari populasi lama)

C1 [0 0 1 1] (hasil crossover)

C2 [1 0 0 1] (hasil crossover)

[0 0 1 0] (hasil mutasi dari C1)

Lalu lakukan Iterasi hingga kriteria tertentu terpenuhi.

Dalam kasus ini, nilai maksimum sebenarnya dapat dihitung secara analitis: $dy/dx = -x + 4 = 0$ $x = 4$ $f(4) = -(4^2)/2 + 4(4) + 40 = 48$.. Nilai maksimum fungsi adalah 48 saat $x = 4$

