Kayla V. Maheswari (2406402454)

No
Date

A. Non - Linear Lagrange Method

* Tujuan : mencari nilai optimum (max / min) suatu fungsi non - linear dengan constraint

* Formula :
  - max / min = $f(x, y)$
  - constrain = $g(x, y) = 0$
  - Lagrangian = $\mathcal{L}(x, y, \lambda) = f(x, y) - \lambda [g(x, y)]$

* Langkah singkat :
  1.) bentuk fungsi lagragian
  2.) turunkan parsial terhadap x, y, dan λ
  3.) set ketiga turunan = 0 (sistem persamaan)
  4.) selesaikan untuk mendapatkan nilai optimum
  5.) cek nilai fungsi pada titik yang diperoleh

B. Markov Chain - Decision Tree

* Tujuan = menguji keputusan terbaik berdasarkan state transition dari markov chain

* Konsep dasar :
  - setiap state memiliki probabilitas transisi
  - Decision tree menambah keputusan sebelum transisi

* Langkah singkat :
  1.) identifikasi state & probabilitas transisi
  2.) Bangun decision tree : node keputusan → node probabilitas
  3.) Hitung expected value tiap cabang
     $E(V) = \Sigma P(x_i) x$
  4.) pilih cabang dengan E(V) tertinggi / cost terendah

C. Markov Chain - Matrix Multiplication

* Tujuan = menganalisis probabilitas jangka panjang / distribusi state kedepan

* Formula :
  $S_n = S_0 \times P^n$
  P = transition matrix
  S_0 = initial state vector

JOYKO 30 Lines, 8 mm

No
Date

* Langkah singkat
  1.) susun transition matrix P
  2.) tentukan vektor awal (S_0)
  3.) lakukan perkalian matrix berulang
  $S_1 = S_0 (P)$
  $S_2 = S_0 (P)^2$
  4.) cari steady state
  $\pi = \pi P$
  $\Sigma \pi_i = 1$

D. Dynamic Knapsack
* Tujuan = menentukan kombinasi barang dengan nilai besar tanpa melebihi kapasitas
* Formula =
  $V(i, w) = \max [V(i-1, w), V(i-1, w-w_i) + v_i]$
  i = index item w_i = berat item
  w = kapasitas v_i = nilai item
* Langkah singkat
  1.) Buat tabel DP ukuran (jumlah item +1) x (kapasitas +1)
  2.) isi baris demi baris menggunakan formula
  3.) Nilai optimal = sel terakhir tabel
  4.) Traceback untuk menentukan item terpilih

E. Queuing type identification
* Tujuan = mengidentifikasi tipe sistem antrean berdasarkan karakteristiknya
* komponen identifikasi
  A / S / C / K / N / D
* Jenis yang diminta
  1.) single normal server (M/M/1)
  - 1 server
  - kedatangan poisson, service exponential
  - Antrean tak terbatas

No
Date

2.) Finite calling population
- Jumlah pelanggan terbatas (mesin 10 → 10 source)
- model : M/M/1/N - source
- Rumus steady state berubah karena arrival rate tergantung jumlah pelanggan tersisa

3.) Finite Queue length (M/M/1/K)
- kapasitas sistem terbatas K (termasuk yang sedang dilayani)
- Jika penuh → Pelanggan ditolak (blocking)

4.) Multiserver (M/M/C)
- lebih dari 1 server (c servers)
- digunakan rumus Erlang C untuk menghitung waiting probability

$$P_w = \frac{(cp)^c / c!(1-p)}{\sum \frac{(cp)^c}{n!} + \frac{(cp)^c}{c!(1-p)}}$$

$$p = \frac{\lambda}{c\mu^*}$$

F. Montecarlo simulation, Queuing system, Probability distribution
* Tujuan = mengsimulasikan proses acak untuk memperkirakan output
* Langkah montecarlo
  1.) Tentukan variable acak & distribusi probabilitasnya
  2.) buat table cumulative probability (CDF)
  3.) Generate random numbers (0-1)
  4.) Cocokan random number dengan interval CDF untuk menentukan outcome
  5.) ulangi ribuan kali untuk estimasi rata-rata, total cost, waktu tunggu, dsb.

* Queuing system
  ↳ • interarrival time (distribution poisson / exponential)
  • service time
  • queue discipline (FIFO)
  • numbers of servers

* Formula
  - interarrival time
    $t = -\frac{1}{\lambda} \ln(1-r)$
  - service time
    $t = -\frac{1}{\mu} \ln(1-r)$
* langkah - langkah
  - masukan waktu kedatangan
  - masukan waktu pelayanan
  - Hitung waktu mulai, dilayani, selesai, dan waktu tunggu
  - ulang untuk banyak pelanggan
  - Hitung rata - rata

* Probability distribution
  - Poisson → $P(X=k) = \frac{e^{-\lambda} \cdot \lambda^{k}}{k!}$
  - eksponential : $\mu e - \mu t$
  - uniform : $x = a + (b-a) r$