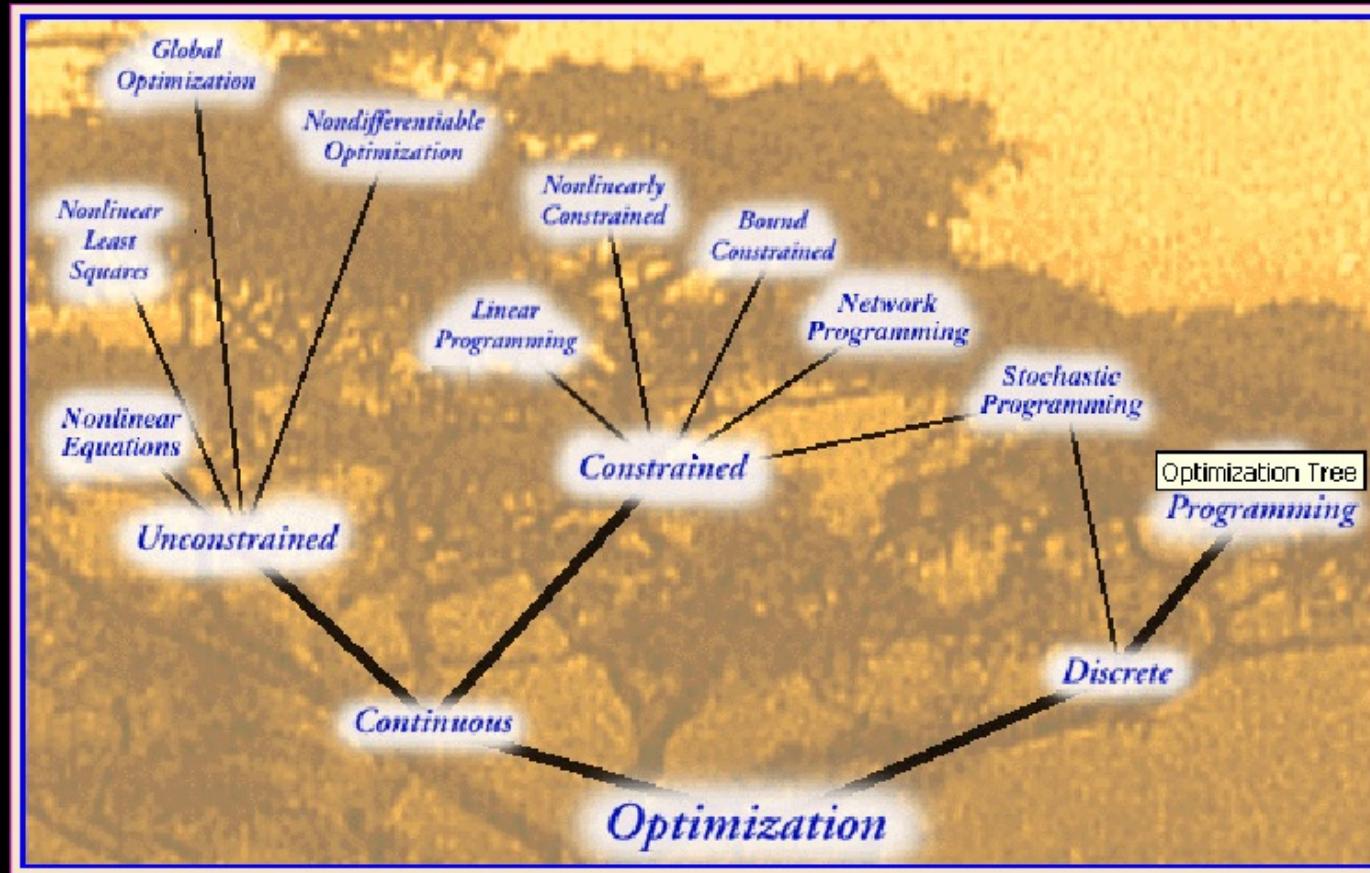
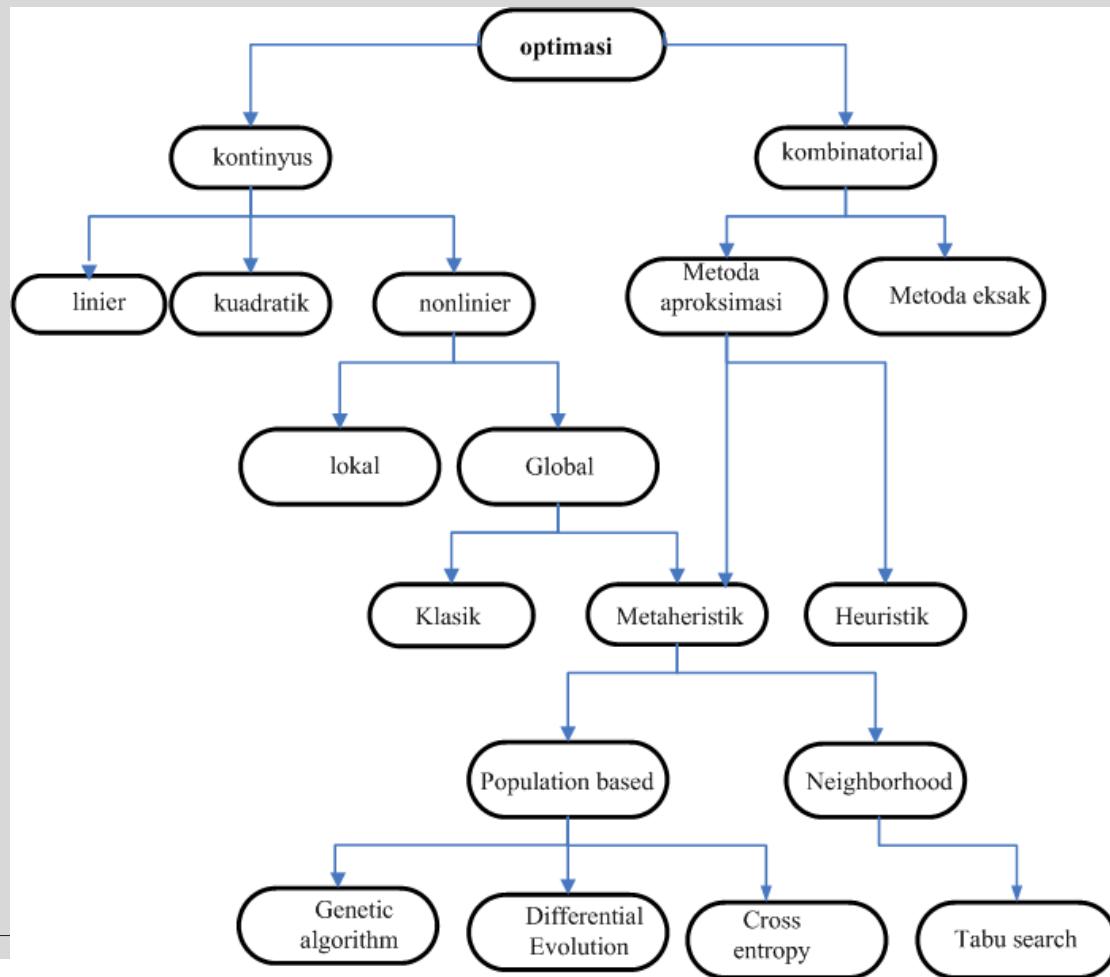


CLUSTERING DENGAN CAT SWARM OPTIMIZATION

Budi Santosa
Teknik Industri ITS
18/2/2022



Klasifikasi Problem optimasi



Masalah Optimasi

- Optimasi tanpa pembatas (unconstrained)
- $\text{Min } f(x) = x^3 - 2x^2 - x + 3, \quad x \in R,$
- $\text{Min } f(x) = x_1^2 - x_2^2 - x_1 + 3, \quad x \in R^2$
- Optimasi dengan pembatas (constrained)

$$\text{Minimasi } f(x) = 2x_1^2 + 3x_2$$

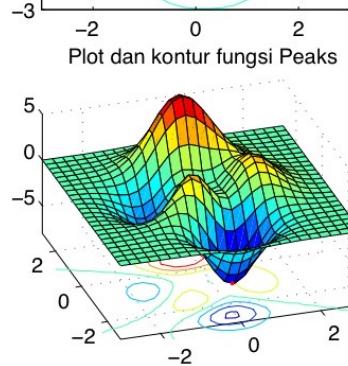
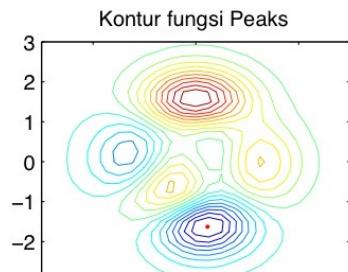
subject to

$$x_1 x_2 \geq 12$$

$$2x_1^2 + 5x_2 \geq 18$$

Local dan global optima

Plot dan kontur fungsi Peaks



$$f = 3(1 - x_1)^2 e^{(-x_1^2 - (x_2 + 1)^2)} - 10 \left(\frac{x_1}{5} - x_1^3 - x_2^5 \right) e^{(-x_1^2 - x_2^2)} - \frac{1}{3} e^{(-(x_1 + 1)^2 - x_2^2)}$$

Optimasi Kombinatorial

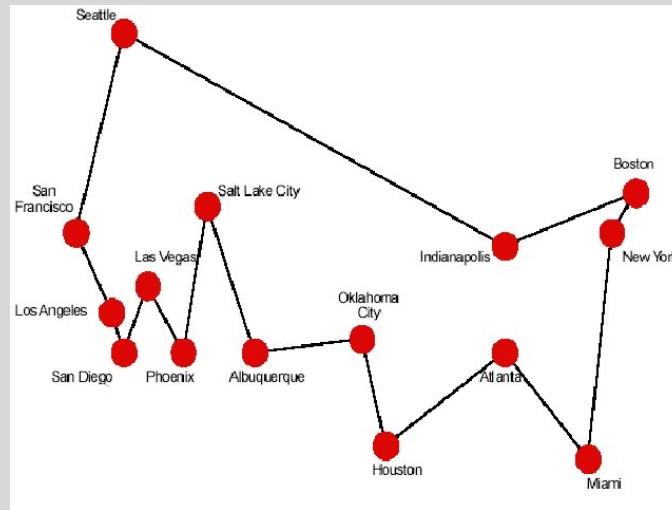
- Traveling Salesman problem
- Penugasan gate untuk pesawat
- Penjadwalan mesin
- Penjadwalan crew pesawat
- Vehicle routing problem
- Multi-product Inventory Ship Routing problem
- General Orientering Problem
- Penjadwalan kuliah/roostering

Travelling Salesman Problem (TSP)

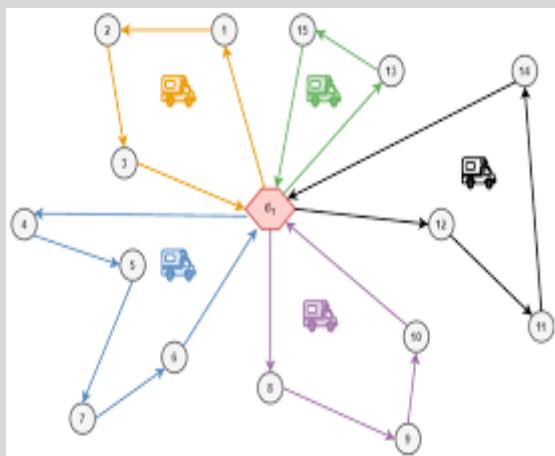
TSP PROBLEM : diberikan N kota, dan matriks jarak antar kota, temukan sebuah rute yang:

- 1. Mengunjungi setiap kota sekali dan hanya sekali**
- 2. Meminimasi jarak total**

- Problem is NP-hard
- Problem optimasi klasik



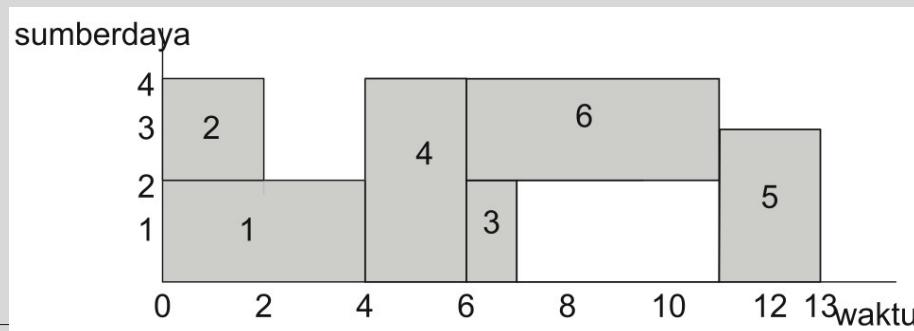
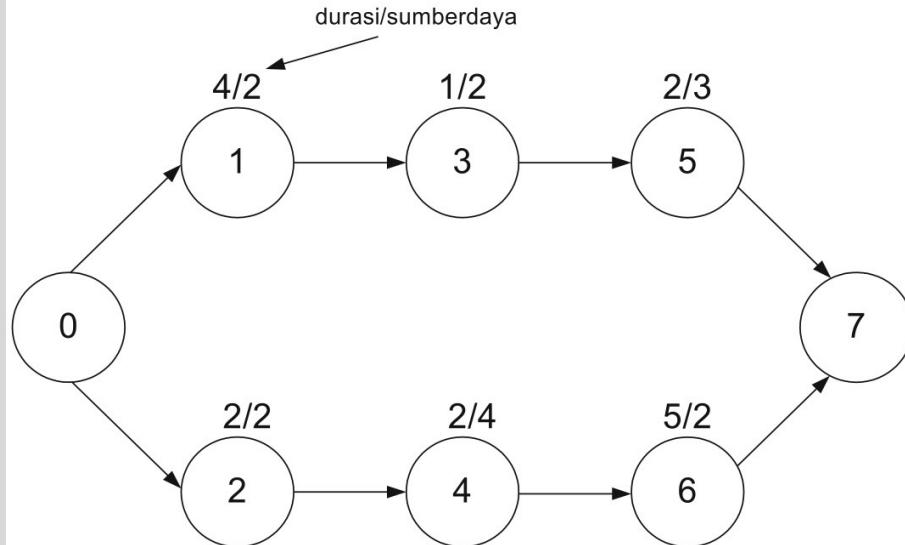
Vehicle Routing Problem



Permasalahan ini mempunyai tujuan untuk menentukan rute perjalanan dari masing-masing kendaraan, dengan ketentuan:

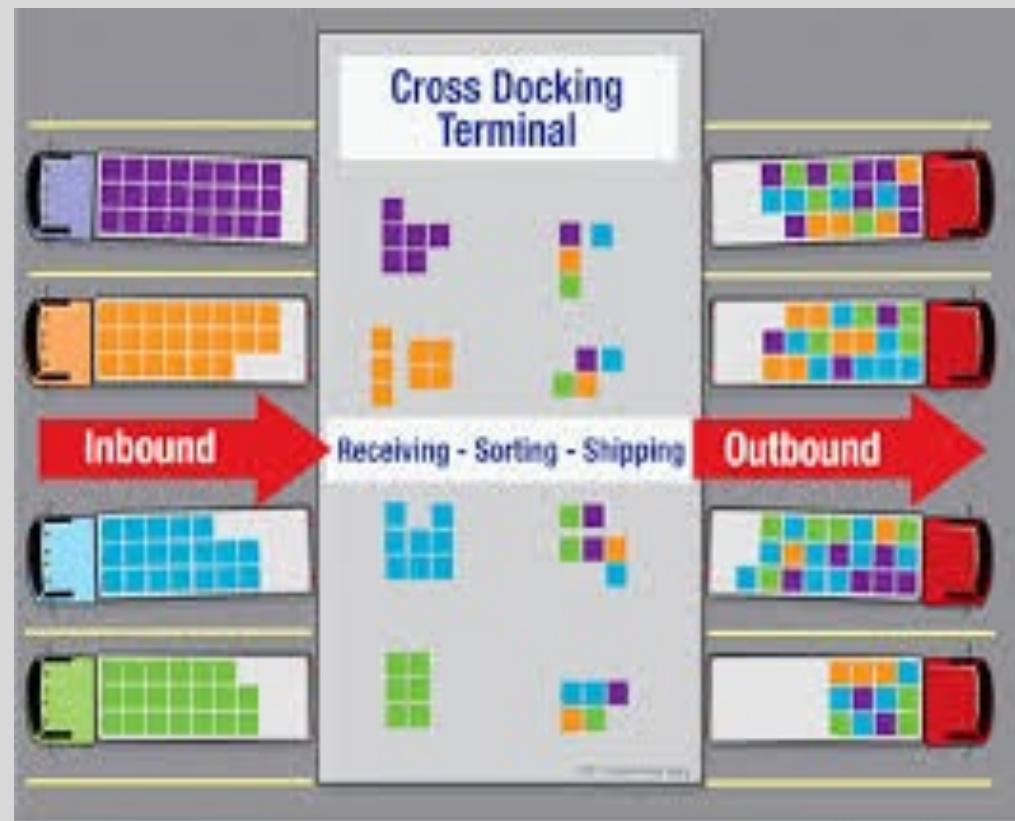
- (1) setiap rute berangkat dari depo dan kembali ke depo,
- (2) setiap pelanggan dikunjungi tepat satu kali oleh hanya satu kendaraan,
- (3) total barang yang dikirim ke pelanggan untuk setiap rute tidak melebihi dari kapasitas angkut kendaraan,
- (4) total waktu operasi masing-masing kendaraan, sudah termasuk waktu transportasi dan waktu pelayanan, tidak melebihi waktu operasi yang diijinkan, dan
- (5) total ongkos keseluruhan rute diminimumkan.

Penjadwalan proyek dengan sumberdaya terbatas atau
Resource-Constrained Project Scheduling Problem (RCPSP)



Menjadwalkan n jobs pada 1 mesin dengan tujuan meminimasi
Total keterlambatan (single machine with minimum total weighted tardiness)

No	L _j	d _j	w _j
1	12	42	7
2	13	33	9
3	14	51	5
4	16	48	14
5	26	63	10
6	31	88	11
7	32	146	8



Clustering

Mengelompokkan obyek ke dalam klaster, dimana setiap klaster berisi obyek yang saling mirip

Kemiripan biasanya diwakili dengan jarak

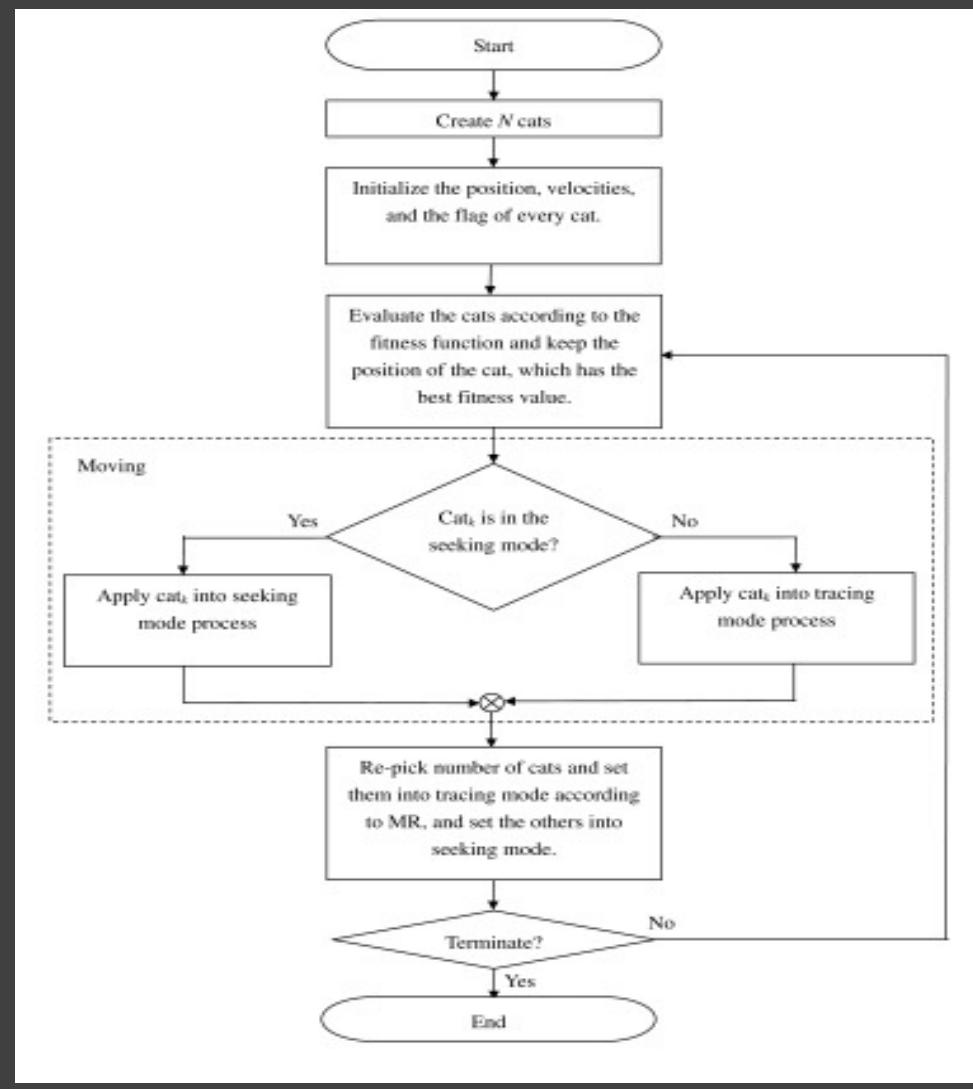
Ada proses optimasi yaitu meminimasi total Sum Squared Error (SSE) . Yaitu, total jarak masing-masing obyek ke pusat klaster

Aplikasi: segmentasi customer, pengelompokan penduduk, pengelompokan produk

Cat Swarm Optimization (CSO)

- Swarm Optimization berdasarkan perilaku pergerakan kucing
- Seeking mode: istirahat tapi tetap dalam kondisi siap memangsa
- Tracing mode: menyergap mangsanya
- Atribut dari setiap kucing/cat
 - Posisi (x)
 - Velocity (v)
- Setiap individual/kucing mempunyai *best position*

ALGORITMA CSO



Seeking Mode

- Seeking mode ini memiliki empat faktor antara lain:
 - *seeking memory pool* (SMP), digunakan untuk menentukan jumlah solusi acak yang dibangkitkan
 - *seeking range of the selected dimension* (SRD) digunakan untuk menentukanberapa jauh posisi akan berubah
 - *counts of dimension to change* (CDC) , dimensi mana yang akan berubah, dan
 - self position consideration (SPC), SPC merupakan nilai bolean (0 dan 1) untuk menentukan apakah posisi dari kucing harus diubah atau dipertahankan.
- Langkah seeking mode intinya adalah bagaimana membuat variasi solusi yang baik

Tracing Mode

- Mengeksekusi mangsa
- Ini langkah lanjutan setelah melakukan seeking model.
- Eksekusi terhadap mangsa dilakukan dengan cara yang cermat.
- Untuk mengimplementasi langkah ini, tracing menggunakan formula yang mirip dengan apa yang ada pada Particle Swarm Optimization (PSO).
- Pada tahap ini solusi yang diciptakan (posisi) akan diupdate berdasarkan kecepatan sebelumnya.
- $V_i^{k+1} = \rho * V_i^k + c_1 r_1 x (p_{besti} - x_i^k) + c_2 x r_2 x (G_{best} - x_i^k)$
- v_i^k : velocity dari setiap particle i pada iterasi k
 c_j : bobot untuk aspek kognitif dan sosial $r1, r2$ =bilangan random antara 0 dan 1, ρ : inersia
- x_i^k : posisi particle i pada iterasi k, p_{besti} :best position dari particle i, G_{best} : best position untuk swarm

Key point metaheuristik

Awalnya CSO untuk meminimasi fungsi kontinyus

Lalu diterapkan pada kasus clustering

Fungsi obyektif yang coba diminimasi adalah SSE (sum squared- error)

Tantangan penerapan metaheuristic adalah

- menentukan struktur solusi
- Mengukur fitness dari solusi

Seeking Mode

1. Define the parameter of seeking mode (SMP, SRD, SPC)

2. For $i = 1$ to k (number of cluster center), do

- Copy cluster center(i) position as many as SMP
- Determine j value
- Calculate the shifting value ($SRD * \text{cluster center}(i)$)

3. For $m = 1$ to SMP, do

- Randomly plus or minus cluster centers with shifting value.
- */The output will be $(SMP \times k)$ cluster center candidates/*

4. Calculate the distance, grouping data into clusters, and calculate SSE

5. Choose a candidate to be the new cluster center roulette wheel selection

Tracing Mode

1. For $i = 1$ to k ,

- Update velocity(i)
- Update position(i),
- get the new cluster center(i)
- $v_{k,d} = w \times v_{k,d} + r \times c \times (x_{best,d} - x_{k,d})$
- $x_{k,d} = v_{k,d} + x$

2. Calculate the distance,
grouping data into clusters, and
calculate SSE

CSO Clustering Algorithm

- Until stopping criteria is met do 1 to 8
- 1. Define the population of data, number of cluster (k), and number of copy
- 2. Choose k data as initial cluster center
- 3. Grouping data into cluster by their closeness, and calculate SSE
- 4. Initialize CSO parameter
- 5. Enter seeking mode
- 6. Compare seeking SSE with earlier SSE.
 - if seeking SSE < earlier SSE use new cluster center.
 - Conversely, use the previous cluster center
- 7. Enter tracing mode
- 8. Compare tracing SSE with earlier SSE.
 - if tracing SSE < earlier SSE use new cluster center.
 - Conversely, use the previous cluster center
- 9. Get SSE, cluster center, and best cluster of each data

Contoh hasil

TABLE I. EXPERIMENTS' DATASET

Dataset	Number of data	Number of attribute	Number of Class
Iris	150	4	3
Soybean-Small	47	35	4
Glass	214	8	6
Balance scale	625	4	3

TABLE V. RESULT OF CLUSTERING FOR DIFFERENT METHODS

	Data	Method	Time	Error
1 Iris		CSO Clustering	0.64	17.55%
		K-means	0.00	18.30%
		PSO Clustering	0.13	27.43%
2 Soybean		CSO Clustering	0.60	24.19%
		K-means	0.00	27.62%
		PSO Clustering	0.22	35.17%
3 Glass		CSO Clustering	1.24	51.63%
		K-means	0.03	48.99%
		PSO Clustering	0.20	56.36%
4 Balance Scale		CSO Clustering	2.59	46.80%
		K-means	0.13	48.11%
		PSO Clustering	0.49	48.74%

Modifikasi

The removed of mixture ratio(MR), so every cat will have to pass the seeking and tracing mode.

This modification is intended to decrease the time needed to find the best cluster center.

Counts of dimension to change (CDC) in seeking mode were assumed always has 100% value, so every dimension of cat's copy will change

Hasil modifikasi

TABLE III. CPU TIME AND CLUSTERING ERROR OF NON-MODIFIED AND MODIFIED CSO CLUSTERING

Dataset	Criteria	CPU time (second)		Clustering Error	
		Mean	std. deviation	Mean (%)	std. deviation
Iris	Original	0.45	0.07	23.34	.119
	Modified	0.64	0.12	17.55	.096
Soybean	Original	0.32	0.04	33.19	.126
	Modified	0.40	0.11	24.60	.094
Glass	Original	1.17	0.22	56.34	.060
	Modified	1.24	0.18	51.63	.049
Balance	Original	2.53	0.72	48.54	.071
Scale	Modified	2.59	0.54	46.80	.060

Kesimpulan

- CSO untuk menyelesaikan persoalan optimasi kontinyus
- Clustering adalah persoalan optimasi
- CSO bisa digunakan untuk menyelesaikan persoalan clustering
- Hasilnya lebih baik dibanding beberapa metoda lain *in terms of SSE*, namun dari sisi waktu komputasi lebih panjang

referensi

- Cat Swarm Optimization, Shu-Chuan Chu , Pei-wei Tsai , and Jeng-Shyang Pan, Q. Yang and G. Webb (Eds.): PRICAI 2006, LNAI 4099, pp. 854 - 858, Springer-Verlag Berlin Heidelberg 2006
- Cat Swarm Optimization for Clustering Budi Santosa dan Mirsa Kencana Ningrum, 2009 International Conference of Soft Computing and Pattern Recognition
- Metoda Metaheuristik, Budi Santosa, Paul willy, Guna Widya 2011
- <https://mti.binus.ac.id/2016/02/23/cat-swarm-optimization/>
-