

Universitatea Tehnică a Moldovei

Programul de master

Știința Datelor

Modele matematice și optimizări

Raport Laborator 2

Elaborat:

masterandul gr. ȘD-241M Sîrbu Valentina

Chișinău, 2024

Problema 1: Transportul produselor între depozite și puncte de distribuție

Date concrete

- Numărul de depozite (m): 5
- Numărul de puncte de distribuție (n): 6
- Costurile de transport (c_{ij}) în unități monetare:

Matricea costurilor:

$$c = \begin{bmatrix} 2 & 3 & 1 & 4 & 5 & 6 \\ 4 & 1 & 3 & 2 & 6 & 5 \\ 3 & 5 & 2 & 6 & 4 & 1 \\ 5 & 6 & 4 & 3 & 2 & 1 \\ 1 & 2 & 3 & 5 & 4 & 6 \end{bmatrix}$$

- Capacitatea depozitelor (s):

$$s = [50, 60, 40, 70, 80]$$

- Cererea punctelor de distribuție (d):

$$d = [40, 50, 60, 70, 50, 30]$$

Cost minim total

Costul minim total al transportului este:

$$\text{Cost minim total} = 490.0$$

Matricea soluției (cantitățile transportate)

Matricea soluției, care arată cantitățile transportate de la fiecare depozit (i) la fiecare punct de distribuție (j), este următoarea:

$$X = \begin{bmatrix} 0 & 0 & 50 & 0 & 0 & 0 \\ 0 & 10 & 0 & 50 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 30 \\ 0 & 0 & 0 & 20 & 50 & 0 \\ 40 & 40 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Interpretarea liniilor matricei

Fiecare linie a matricei X reprezintă transporturile realizate de la un depozit la punctele de distribuție:

- **Depozitul 1** ($i = 1$): Transportă 50 de unități către punctul de distribuție 3.

```

Cost minim total: 490.0
Matricea soluției (cantitățile transportate):
[[ 0.  0. 50.  0.  0.  0.]
 [ 0. 10.  0. 50.  0.  0.]
 [ 0.  0. 10.  0.  0. 30.]
 [ 0.  0.  0. 20. 50.  0.]
 [40. 40.  0.  0.  0.  0.]]

```

	Distributie	D1	D2	D3	D4	D5	D6		Capacitate	Cost total
Depozite										
D1			2	3	1	4	5	6	50	490
D2			4	1	3	2	6	5	60	
D3			3	5	2	6	4	1	40	
D4			5	6	4	3	2	1	70	
D5			1	2	3	5	4	6	80	
	Cererea		40	50	60	70	50	30	sum= 300	
	Distributie	D1	D2	D3	D4	D5	D6			
Depozite										
D1			0	0	50	0	0	0	50	
D2			0	10	0	50	0	0	60	
D3			0	0	10	0	0	30	40	
D4			0	0	0	20	50	0	70	
D5			40	40	0	0	0	0	80	
	Rezultat		40	50	60	70	50	30		

- **Depozitul 2 ($i = 2$):** Transportă:
 - 10 unități către punctul de distribuție 2,
 - 50 de unități către punctul de distribuție 4.
- **Depozitul 3 ($i = 3$):** Transportă:
 - 10 unități către punctul de distribuție 3,
 - 30 de unități către punctul de distribuție 6.
- **Depozitul 4 ($i = 4$):** Transportă:
 - 20 de unități către punctul de distribuție 4,
 - 50 de unități către punctul de distribuție 5.
- **Depozitul 5 ($i = 5$):** Transportă:
 - 40 de unități către punctul de distribuție 1,
 - 40 de unități către punctul de distribuție 2.

Verificarea constrângerilor

1. Capacitatea depozitelor: Suma cantităților transportate din fiecare depozit este mai mică sau egală cu capacitatea sa:

$$\text{Depozitul 1: } 50 \leq 50,$$

$$\text{Depozitul 2: } 10 + 50 = 60 \leq 60,$$

$$\text{Depozitul 3: } 10 + 30 = 40 \leq 40,$$

$$\text{Depozitul 4: } 20 + 50 = 70 \leq 70,$$

$$\text{Depozitul 5: } 40 + 40 = 80 \leq 80.$$

2. Cererea punctelor de distribuție: Suma cantităților primite de fiecare punct de distribuție este mai mare sau egală cu cererea acestuia:

$$\text{Punctul 1: } 40 \geq 40,$$

$$\text{Punctul 2: } 10 + 40 = 50 \geq 50,$$

$$\text{Punctul 3: } 50 + 10 = 60 \geq 60,$$

$$\text{Punctul 4: } 50 + 20 = 70 \geq 70,$$

$$\text{Punctul 5: } 50 \geq 50,$$

$$\text{Punctul 6: } 30 \geq 30.$$

Worksheet: [MM0tema2.xlsx]Echi

Report Created: 1/22/2025 7:24:00 PM

Result: Solver found a solution. All constraints and optimality conditions are satisfied.

Solver Engine

- **Engine:** Simplex LP
- **Solution Time:** 0.153 Seconds
- **Iterations:** 30 (Subproblem: 0)

Solver Options

- Max Time: Unlimited
- Iterations: Unlimited
- Use Automatic Scaling
- Max Subproblems: Unlimited
- Max Integer Solutions: Unlimited
- Integer Tolerance: 1%
- Assume NonNegative

Cell	Name	Original Value	Final Value
\$M\$4	D1 Cost total	0	490

Objective Cell (Min)

Variable Cells

Cell	Name	Original Value	Final Value	Integer
\$C\$16	D1 D1	0	0	Integer
\$D\$16	D1 D2	0	0	Integer
\$E\$16	D1 D3	0	50	Integer

Constraints

Cell	Name	Cell Value	Formula	Status/Slack
\$C\$22	Rezultat D1	40	\$C\$22=\$C\$11	Binding, Slack=0
\$D\$22	Rezultat D2	50	\$D\$22=\$D\$11	Binding, Slack=0
\$E\$22	Rezultat D3	60	\$E\$22=\$E\$11	Binding, Slack=0

Concluzie

Rezultatele obținute respectă toate constrângerile problemei de transport, iar costul total de transport este minimizat. Aceasta este soluția optimă pentru problema dată.

Problema 2: Transportul produselor între depozite

Date concrete

În această problemă, avem 10 surse și 3 destinații, ceea ce rezultă într-o matrice de costuri 10×3 . Costurile asociate transportului dintre fiecare sursă și destinație sunt date de matricea:

$$\text{Costuri} = \begin{bmatrix} 12 & 8 & 15 \\ 10 & 18 & 9 \\ 14 & 9 & 19 \\ 11 & 12 & 10 \\ 20 & 13 & 11 \\ 17 & 15 & 14 \\ 10 & 16 & 12 \\ 13 & 11 & 15 \\ 18 & 14 & 9 \\ 14 & 10 & 17 \end{bmatrix}$$

Oferta (capacitatea surselor):

[25, 30, 20, 40, 35, 15, 50, 30, 25, 30] unități.

Cererea (destinațiile):

[70, 100, 80] unități.

Problema este neechilibrată deoarece oferta totală (300) depășește cererea totală (250).

Cost minim total

Folosind algoritmul din pachetul `scipy.optimize`, am rezolvat problema pentru a minimiza costul total de transport. Costul minim calculat este:

Cost minim total: **2425.0** unități monetare.

Matricea soluției (cantitățile transportate)

Cantitățile optime transportate între fiecare sursă și destinație sunt organizate în matricea următoare:

$$\text{Matricea soluției} = \begin{bmatrix} 0 & 25 & 0 \\ 0 & 0 & 30 \\ 0 & 20 & 0 \\ 20 & 0 & 20 \\ 0 & 0 & 5 \\ 0 & 0 & 0 \\ 50 & 0 & 0 \\ 0 & 25 & 0 \\ 0 & 0 & 25 \\ 0 & 30 & 0 \end{bmatrix}$$

```
Cost minim total: 2425.0
Matricea soluției (cantitățile transportate):
[[ 0. 25.  0.]
 [ 0.  0. 30.]
 [ 0. 20.  0.]
 [20.  0. 20.]
 [ 0.  0.  5.]
 [ 0.  0.  0.]
 [50.  0.  0.]
 [ 0. 25.  0.]
 [ 0.  0. 25.]
 [ 0. 30.  0.]]
```


- Surplusurile sunt incluse pentru a compensa neechilibrul între ofertă și cerere.

Worksheet: [MM0tema2.xlsx]NoEchi

Report Created: 1/22/2025 7:10:34 PM

Result: Solver found a solution. All constraints and optimality conditions are satisfied.

Solver Engine

- **Engine:** Simplex LP
- **Solution Time:** 0.225 Seconds
- **Iterations:** 33 (Subproblem: 0)

Solver Options

- Max Time: Unlimited
- Iterations: Unlimited
- Use Automatic Scaling
- Max Subproblems: Unlimited
- Max Integer Solutions: Unlimited
- Integer Tolerance: 1%
- Assume NonNegative

Objective Cell (Min)

Cell	Name	Original Value	Final Value
\$K\$5	E1 Cost total	0	2425

Variable Cells

Cell	Name	Original Value	Final Value	Integer
\$C\$23	E1 D1	0	0	Integer
\$D\$23	E1 D2	0	25	Integer
\$E\$23	E1 D3	0	0	Integer
\$C\$24	E2 D1	0	20	Integer
\$D\$24	E2 D2	0	0	Integer
\$E\$24	E2 D3	0	10	Integer

Constraints

Cell	Name	Cell Value	Formula	Status/Slack
\$C\$35	Cerere D1	70	\$C\$35=\$C\$17	Binding, Slack=0
\$D\$35	Cerere D2	100	\$D\$35=\$D\$17	Binding, Slack=0
\$E\$35	Cerere D3	80	\$E\$35=\$E\$17	Binding, Slack=0
\$H\$23	E1 Capacitate	25	\$H\$23 _i =\$H\$5	Binding, Slack=0
\$H\$24	E2 Capacitate	30	\$H\$24 _i =\$H\$6	Binding, Slack=0

Concluzie

Problema de transport neechilibrată a fost rezolvată cu succes, minimizând costul total de transport la **2425.0**. Soluția respectă toate constrângerile impuse, iar distribuția optimă este prezentată în matricea soluției.

Traveling Salesman Problem (TSP)

Setul de date este format din coordonatele bidimensionale (într-un sistem 2D) ale celor 70 de orașe. Tipul distanței utilizate este cel euclidian (EUC_2D).

Structura datelor

- **NAME:** st70
- **TYPE:** TSP
- **DIMENSION:** 70
- **EDGE_WEIGHT_TYPE:** EUC_2D

Coordonatele orașelor

Coordonatele fiecărui oraș sunt prezentate în tabelul de mai jos:

ID	Coordonata X	Coordonata Y
1	64	96
2	80	39
3	69	23
4	72	42
5	48	67
6	58	43
7	81	34
8	79	17
9	30	23

10	42	67
11	7	76
12	29	51
13	78	92
14	64	8
15	95	57
16	57	91
17	40	35
18	68	40
19	92	34
20	62	1
21	28	43
22	76	73
23	67	88
24	93	54
25	6	8
26	87	18
27	30	9
28	77	13
29	78	94
30	55	3
31	82	88
32	73	28
33	20	55
34	27	43
35	95	86
36	67	99
37	48	83
38	75	81
39	8	19
40	20	18
41	54	38
42	63	36
43	44	33
44	52	18
45	12	13
46	25	5
47	58	85
48	5	67
49	90	9
50	41	76
51	25	76
52	37	64
53	56	63
54	10	55
55	98	7

56	16	74
57	89	60
58	48	82
59	81	76
60	29	60
61	17	22
62	5	45
63	79	70
64	9	100
65	17	82
66	74	67
67	10	68
68	48	19
69	83	86
70	84	94

Table 1: Coordonatele orașelor din problema TSP

Pași principali ai codului

1. Importarea bibliotecilor necesare:

- Biblioteci standard precum `iomanip`, `vector`, `random`, etc., sunt importate pentru a facilita funcționalitățile necesare.

2. Structura Point:

- Definește un punct 2D cu coordonatele `x` și `y`.

3. Citirea fișierelor TSP:

- Funcția `read_instances()` citește datele de intrare (`NODE_COORD_SECTION` sau `EDGE_WEIGHT_SECTION`) și construiește matricea distanțelor.
- Dacă datele sunt sub formă de coordonate, distanțele sunt calculate folosind formula distanței euclidiene:

$$d(a, b) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

4. Generarea populației inițiale:

- Funcția `generate_initial_population()` creează o populație inițială de permutări ale orașelor folosind un generator de numere aleatoare.

5. Calcularea costului pentru un individ:

- Funcția `total_dist_individual()` calculează costul total al unui traseu, inclusiv întoarcerea la orașul inițial.

6. Selecția indivizilor:

- Se utilizează metode precum *rank selection* și *tournament selection* pentru a selecta indivizi în funcție de fitness.

7. Recombinarea (Crossover):

- Funcția `crossover()` combină doi părinți pentru a crea doi urmași folosind o secțiune comună și completând cu orașele rămase.

8. Mutarea (Mutation):

- Funcția `mutation()` schimbă pozițiile a două orașe într-un traseu pentru a introduce variație în populație.

9. Refinarea soluțiilor:

- Algoritmul *2-opt* este aplicat la fiecare 10 generații pentru a optimiza local traseele.

10. Rularea algoritmului genetic:

- Funcția principală `run_ga()` execută algoritmul pe parcursul mai multor generații. Fiecare generație constă în:
 - Selectarea indivizilor pentru reproducere.
 - Generarea urmașilor prin recombinare și mutație.
 - Calcularea fitness-ului și sortarea populației.

11. Evaluarea soluției:

- Costul minim din populația finală este afișat ca soluție optimă găsită de algoritm.

```

Generation 1992 - Best Cost: 677.11
Generation 1993 - Best Cost: 677.11
Generation 1994 - Best Cost: 677.11
Generation 1995 - Best Cost: 677.11
Generation 1996 - Best Cost: 677.11
Generation 1997 - Best Cost: 677.11
Generation 1998 - Best Cost: 677.11
Generation 1999 - Best Cost: 677.11
Generation 2000 - Best Cost: 677.11
Final population (best individuals):
Best individual:
Total cost: 677.11
Route: 30 37 58 21 65 62 56 14 23 18 6 1 3 17 41 31 2 7 25 54 48 27 13 19 29 43 67 26 45 44 24 38 60 39 8 16 42 40 5 52
4 9 51 59 11 20 33 32 61 53 47 66 10 63 64 55 50 49 57 36 46 15 0 35 22 12 28 69 34 68

```

Rezultate și concluzii

Algoritmul genetic a obținut o soluție de **677.11**, aproape de soluția optimă globală de **675** (*eroare de 0.31%*). Performanța sa demonstrează eficiență, dar există oportunități de îmbunătățire:

- **Creșterea diversității populației:** Introducerea unor operatori de mutație mai agresivi sau strategii pentru reintroducerea diversității.
- **Optimizare locală mai frecventă:** Aplicarea intensificată a metodei *2-opt* sau utilizarea altor metode de optimizare locală (ex. *3-opt*).
- **Ajustarea parametrilor:** Optimizarea ratei de mutație, dimensiunii populației sau a numărului de generații.
- **Metode hibride:** Combinarea algoritmului genetic cu metode deterministe, cum ar fi algoritmi bazați pe *branch and bound*.

Link soluție optimă:

<http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/STSP.html>