

Doktorantūros ataskaita

Mindaugas Kepalas (IV kursas, I semestras)

2024-09-29

Doktorantūros suvestinė

Doktorantas

Mindaugas Kepalas

Disertacijos tema

Optimalus vietų išdėstymas tinkle

Doktorantūros vadovas

prof. dr. (HP) Julius Žilinskas

Doktorantūros laikotarpis

2020 spalio 1 d. — 2025 kovo 31 d.

Kursas, semestras

IV kursas I semestras (pusmetį buvau akademinėse atostogose)

Studijų plano suvestinė

Studijų metai	Egzaminai		Dalyvavimas konferencijose		Publikacijos		
	Planas	Įvykdyta	Planas	Įvykdyta	Planas	Įvykdyta	Būklė
I (2020/2021)	2	2					
II (2021/2022)	2	2	1	2			
III (2022/2023)			1	1	1	1	Priimta
IV (2023/2024)					1	0	Ruošiama

Ataskaitinio pusmečio suvestinė

Dalyvavimas konferencijose		
Planas	Ivykdyta	Konferencijos tipas
	Mindaugas Kepalas, Julius Žilinkas. “Net-Constrained Clustering Problem”. DAMSS 2023, Druskininkai.	Nacionalinė

Publikacijos			
Planas	Ivykdyta	Būklė	Publikacijos tipas
Publikacija moksliniame žurnale, turinčiame cituojamumo rodiklį	Mindaugas Kepalas, Julius Žilinkas. “Solving Net-Constrained Clustering Problem”. Journal of Nonlinear and Variational Analysis http://inva.biemdas.com/	Priimtas	Su cituojamumo rodikliu

Mokslinių tyrimų ir disertacijos rengimo etapai

Darbo pavadinimas	Atlikimo terminai	Pastabos
Disertacijos tikslų ir uždavinių formulavimas	2021 rugsėjo mėn.	Atlikta
Literatūros apžvalga	Visu doktorantūros metu	Atliekama
Uždavinių teorinis tyrimas	Visu doktorantūros metu	Atlikta
Optimizavimo algoritmų programavimas (uždavinių sprendimas)	Visu doktorantūros metu	Atliekama
Mokslinės literatūros (straipsnių, knygų) skaitymas, sisteminimas, analizė	Visu doktorantūros metu	Atliekama
Dalyvavimas konferencijoje (-ose)	Visu doktorantūros metu	Atlikta
Moksliniai straipsniai	2023 pavasaris (neseniai priimta), 2025 žiema	Pirmas straipsnis priimtas, antras ruošiamas
Disertacijos rašymas	2024 sausio 1 d. - 2024 kovo 31 d.	Planuojama pradėti išspausdinus antrąjį straipsnį
Disertacijos įteikimas	2024 balandžio mėn.	

Gauti moksliniai rezultatai ir planai kitam semestru

IV kurso I semestre gauti moksliniai rezultatai: žr. mokslinę ataskaitą, pirmas straipsnis priimtas, antras straipsnis rengiamas ir derinamas su vadovu.

Planai IV kurso II semestru:

- 1.** Parengti antrąjį straipsnį ir įteikti žurnalui (iki 2024 gruodžio 31 d.)
- 2.** Parašyti disertaciją (iki 2025 kovo 31 d.)

Straipsnio apžvalga: uždavinio formuluotė

Straipsnio apžvalga: uždavinio formuluotė

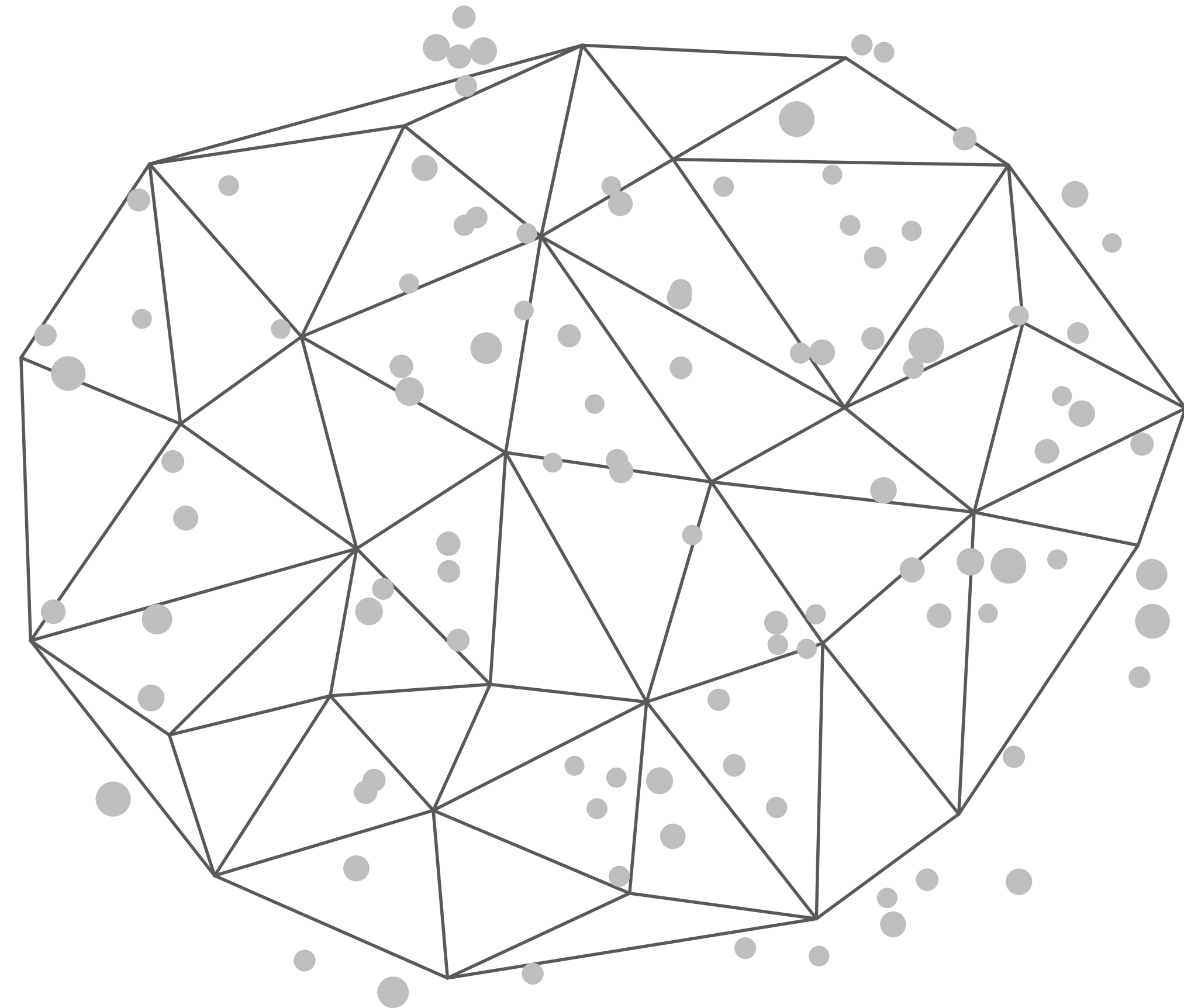
- **Duota:**

- Taškų aibė $\{P_1, P_2, \dots, P_N\}$ su svoriais $\{w_1, w_2, \dots, w_N\}$
- “Tinklas” $\mathcal{N}$ (šis “tinklas” yra M segmentų sąjunga)
- Klasterių skaičius K

Straipsnio apžvalga: uždavinio formuluotė

- **Duota:**

- Taškų aibė $\{P_1, P_2, \dots, P_N\}$ su svoriais $\{w_1, w_2, \dots, w_N\}$
- “Tinklas” $\mathcal{N}$ (šis “tinklas” yra M segmentų sąjunga)
- Klasterių skaičius K



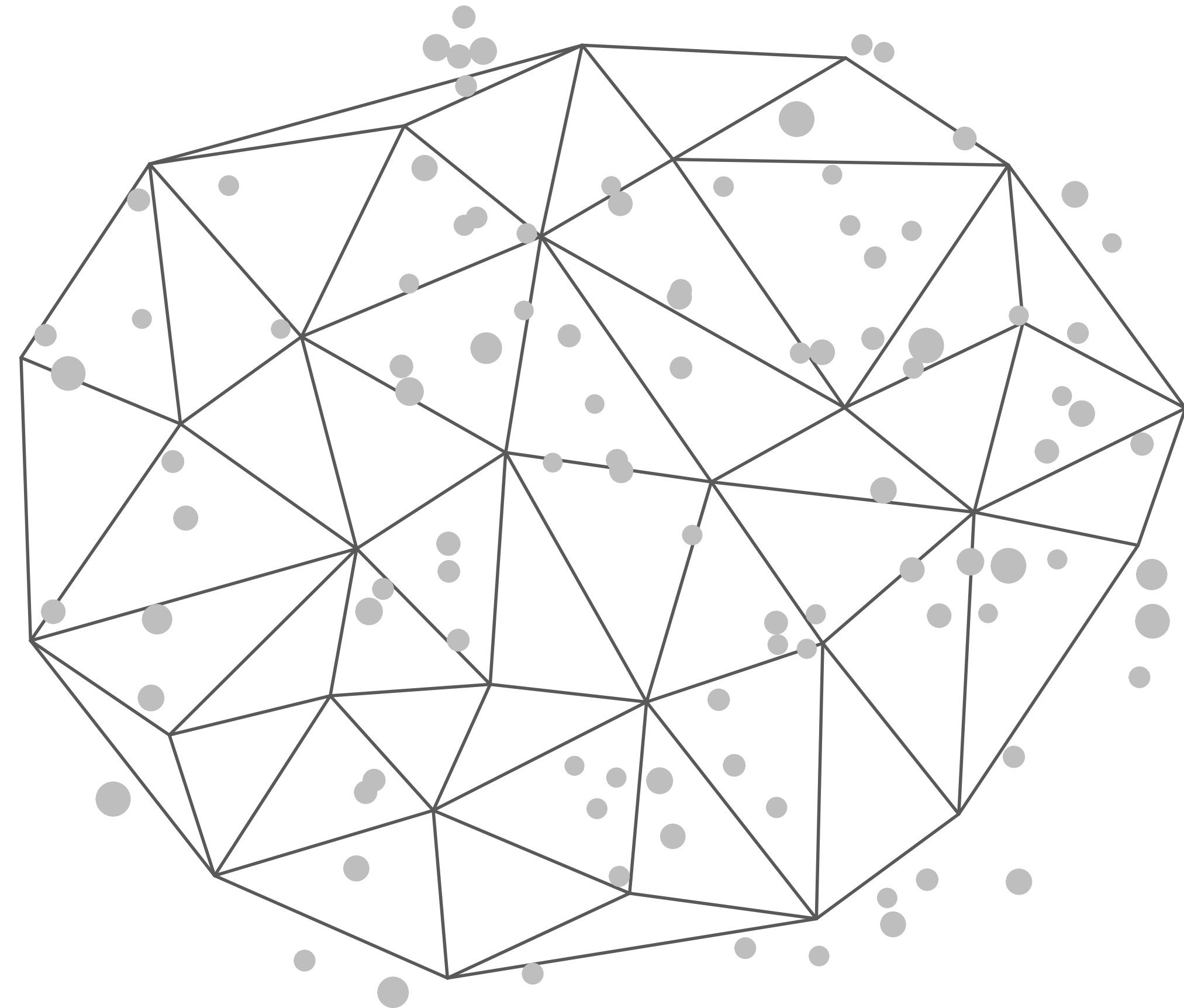
Straipsnio apžvalga: uždavinio formuluotė

- **Duota:**

- Taškų aibė $\{P_1, P_2, \dots, P_N\}$ su svoriais $\{w_1, w_2, \dots, w_N\}$
- “Tinklas” $\mathcal{N}$ (šis “tinklas” yra M segmentų sąjunga)
- Klasterių skaičius K

- **Uždavinys.** Apibrėžti K klasterių $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_K$ su atitinkamais centrais $Q_1, Q_2, \dots, Q_K$, sprendžiančių uždavinį

$$\min_{Q_1, Q_2, \dots, Q_K} \sum_{k=1}^K \sum_{i \in \mathcal{C}_k} w_i \left\| P_i - Q_k \right\|_2^2 \quad \text{s.t.} \quad Q_k \in \mathcal{N}, k = 1, \dots, K$$



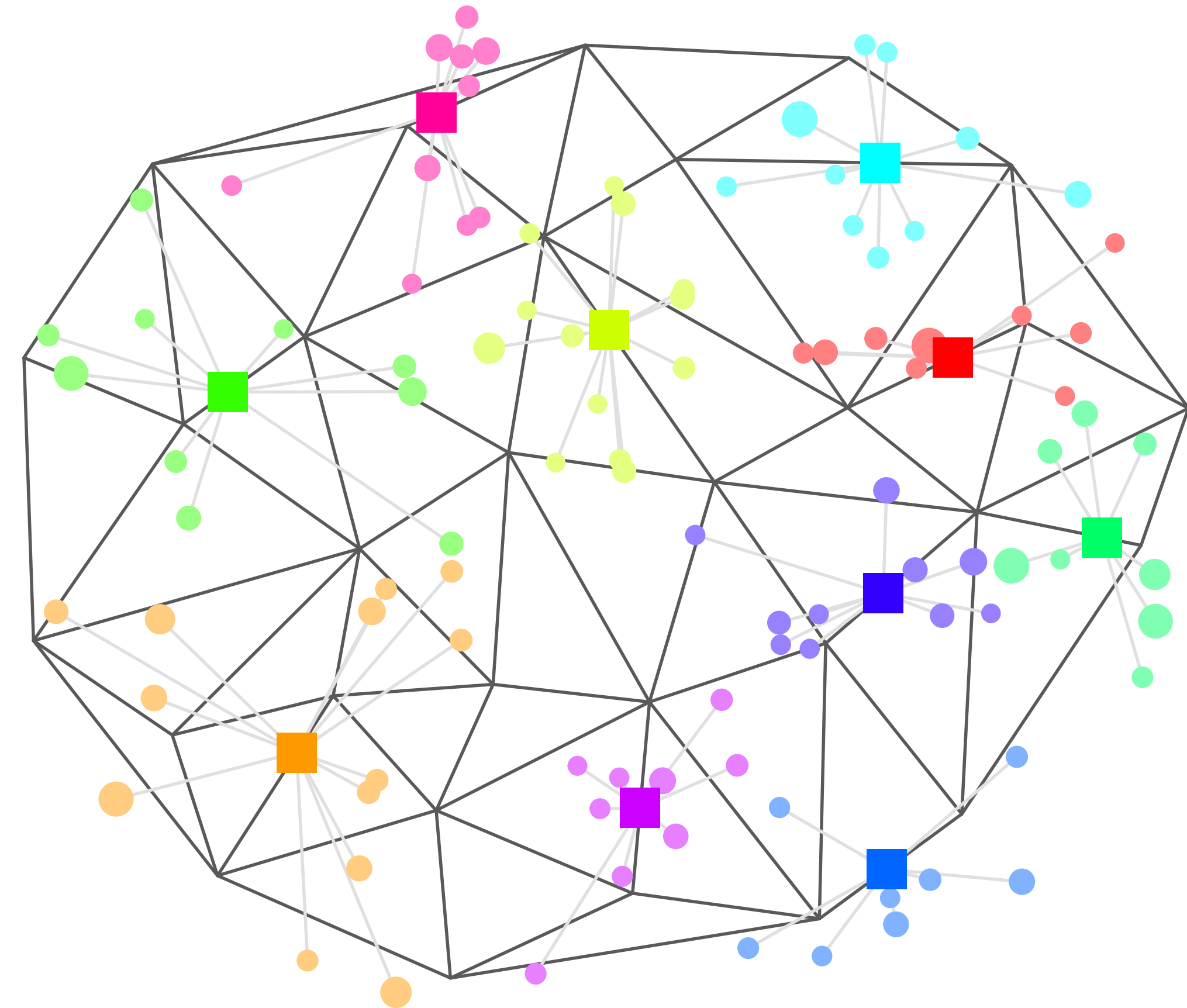
Straipsnio apžvalga: uždavinio formuluotė

- **Duota:**

- Taškų aibė $\{P_1, P_2, \dots, P_N\}$ su svoriais $\{w_1, w_2, \dots, w_N\}$
- “Tinklas” $\mathcal{N}$ (šis “tinklas” yra M segmentų sąjunga)
- Klasterių skaičius K

- **Uždavinys.** Apibrėžti K klasterių $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_K$ su atitinkamais centrais $Q_1, Q_2, \dots, Q_K$, sprendžiančių uždavinį

$$\min_{Q_1, Q_2, \dots, Q_K} \sum_{k=1}^K \sum_{i \in \mathcal{C}_k} w_i \left\| P_i - Q_k \right\|_2^2 \quad \text{s.t.} \quad Q_k \in \mathcal{N}, k = 1, \dots, K$$



Straipsnio apžvalga: uždavinio formuluotė

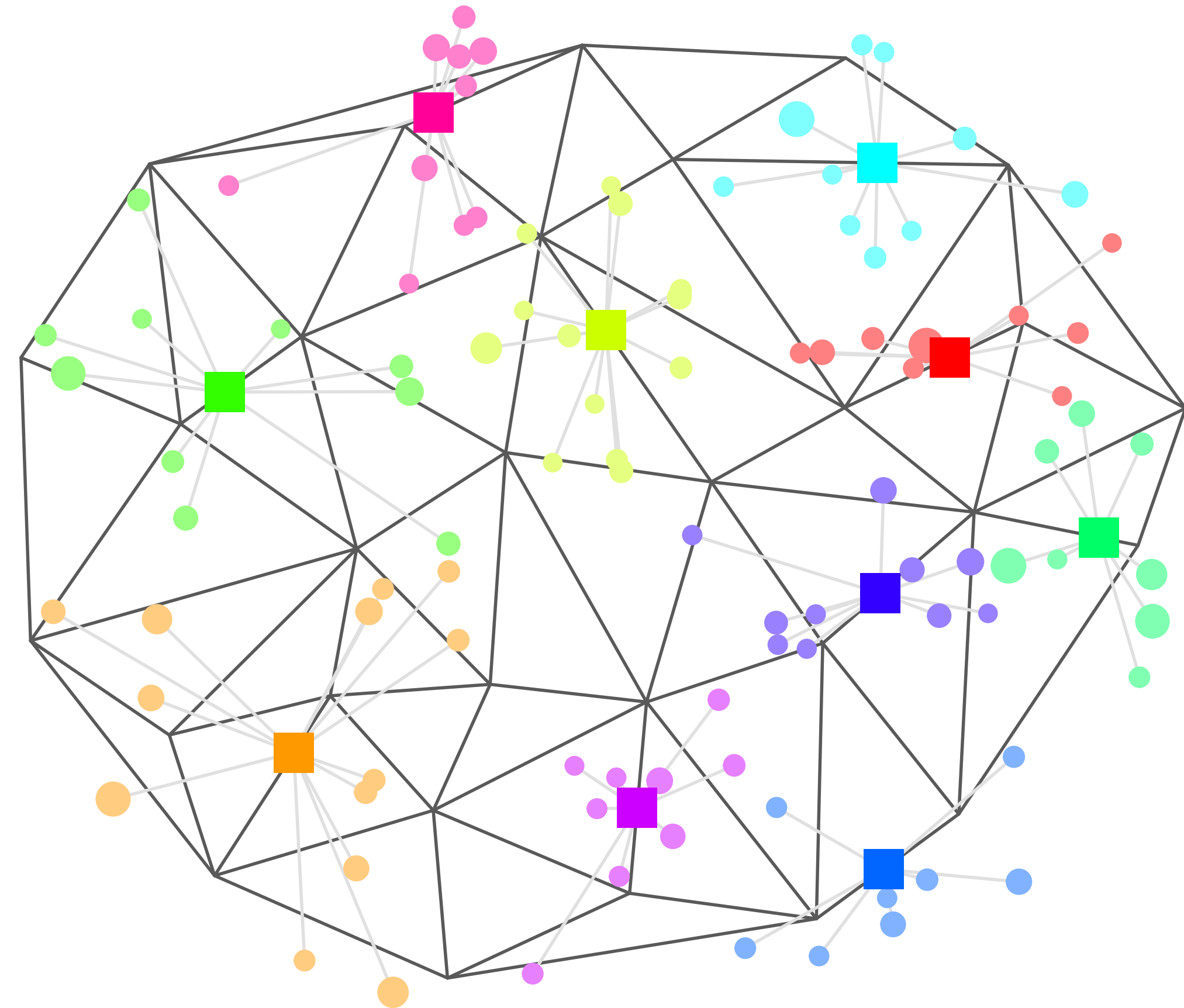
- **Duota:**

- Taškų aibė $\{P_1, P_2, \dots, P_N\}$ su svoriais $\{w_1, w_2, \dots, w_N\}$
- “Tinklas” $\mathcal{N}$ (šis “tinklas” yra M segmentų sąjunga)
- Klasterių skaičius K

- **Uždavinys.** Apibrėžti K klasterių $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_K$ su atitinkamais centrais $Q_1, Q_2, \dots, Q_K$, sprendžiančių uždavinį

$$\min_{Q_1, Q_2, \dots, Q_K} \sum_{k=1}^K \sum_{i \in \mathcal{C}_k} w_i \left\| P_i - Q_k \right\|_2^2 \quad \text{s.t.} \quad Q_k \in \mathcal{N}, k = 1, \dots, K$$

- Atkreipkite dėmesį į tinklo ribojimą klasterių centrams (kvadratėliams): $Q_k \in \mathcal{N}$



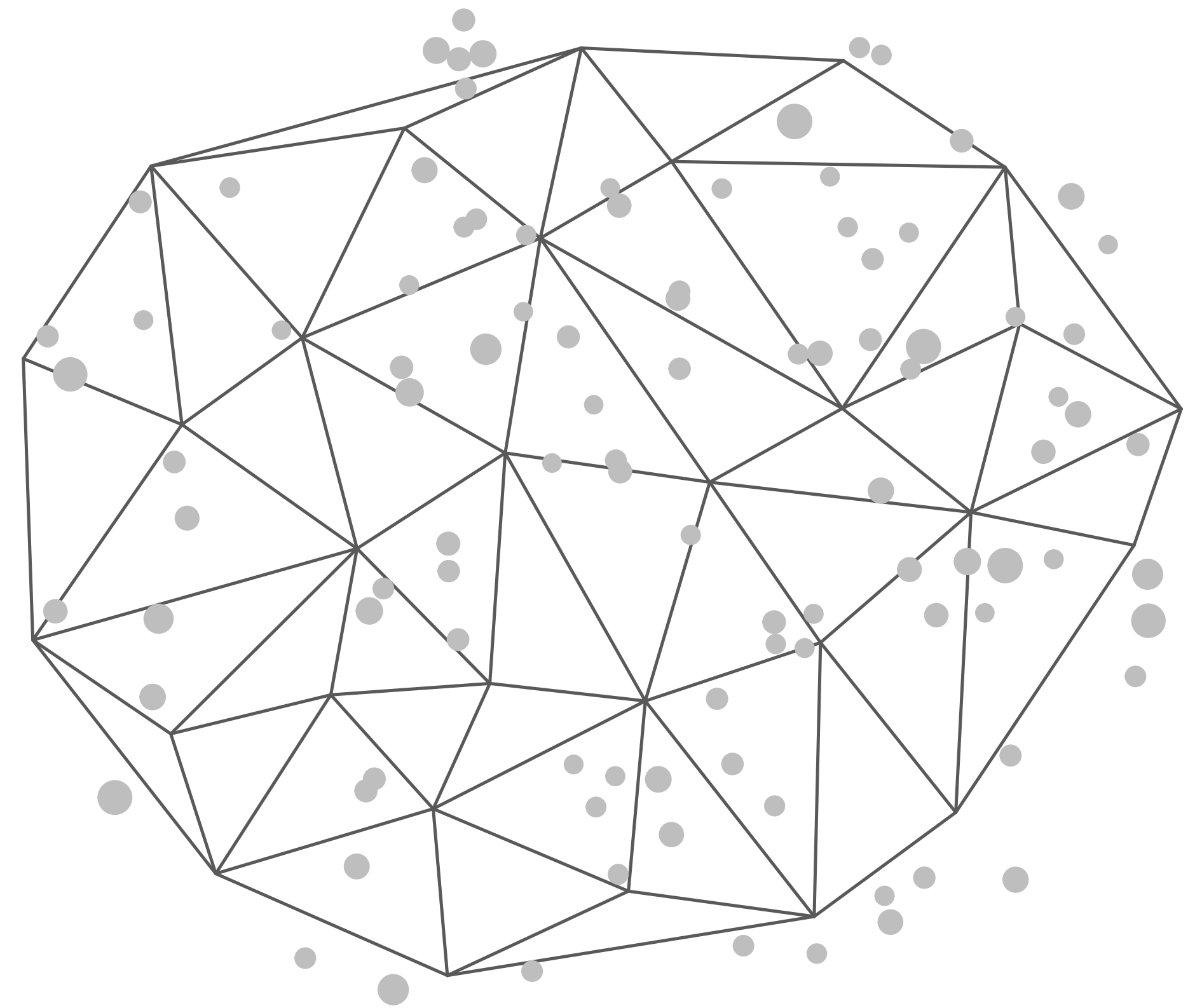
Straipsnio apžvalga: net-k-means algoritmas

Straipsnio apžvalga: net-k-means algoritmas

- **Net-k-means algoritmas** skirtas rasti lokaliu klasteriavimo uždavinio su tinklo ribojimu centrams. **Šablonas:**

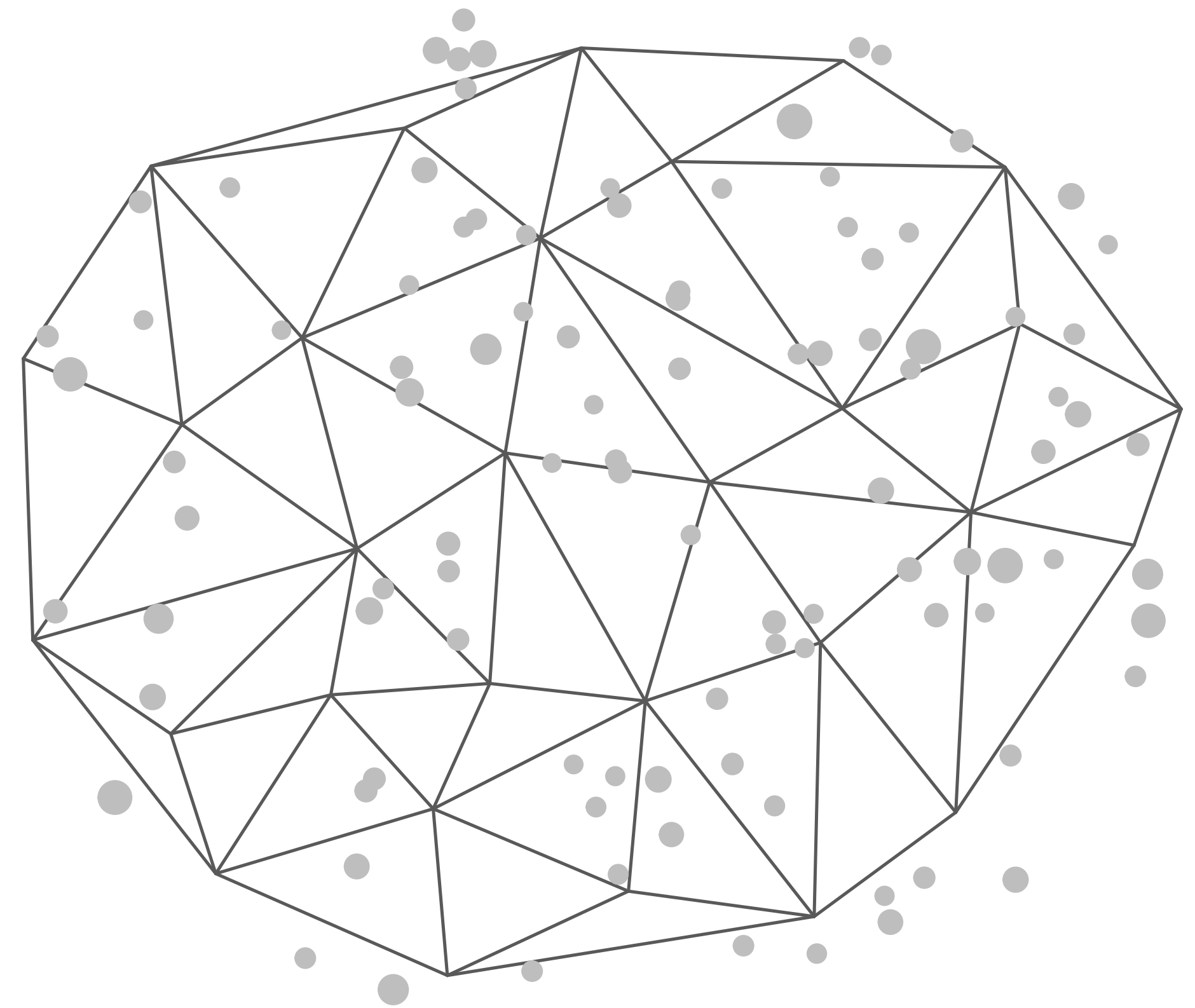
Straipsnio apžvalga: net-k-means algoritmas

- **Net-k-means algoritmas** skirtas rasti lokalią klasteriavimo užduotį su tinklo ribojimu centrams. **Šablonas:**



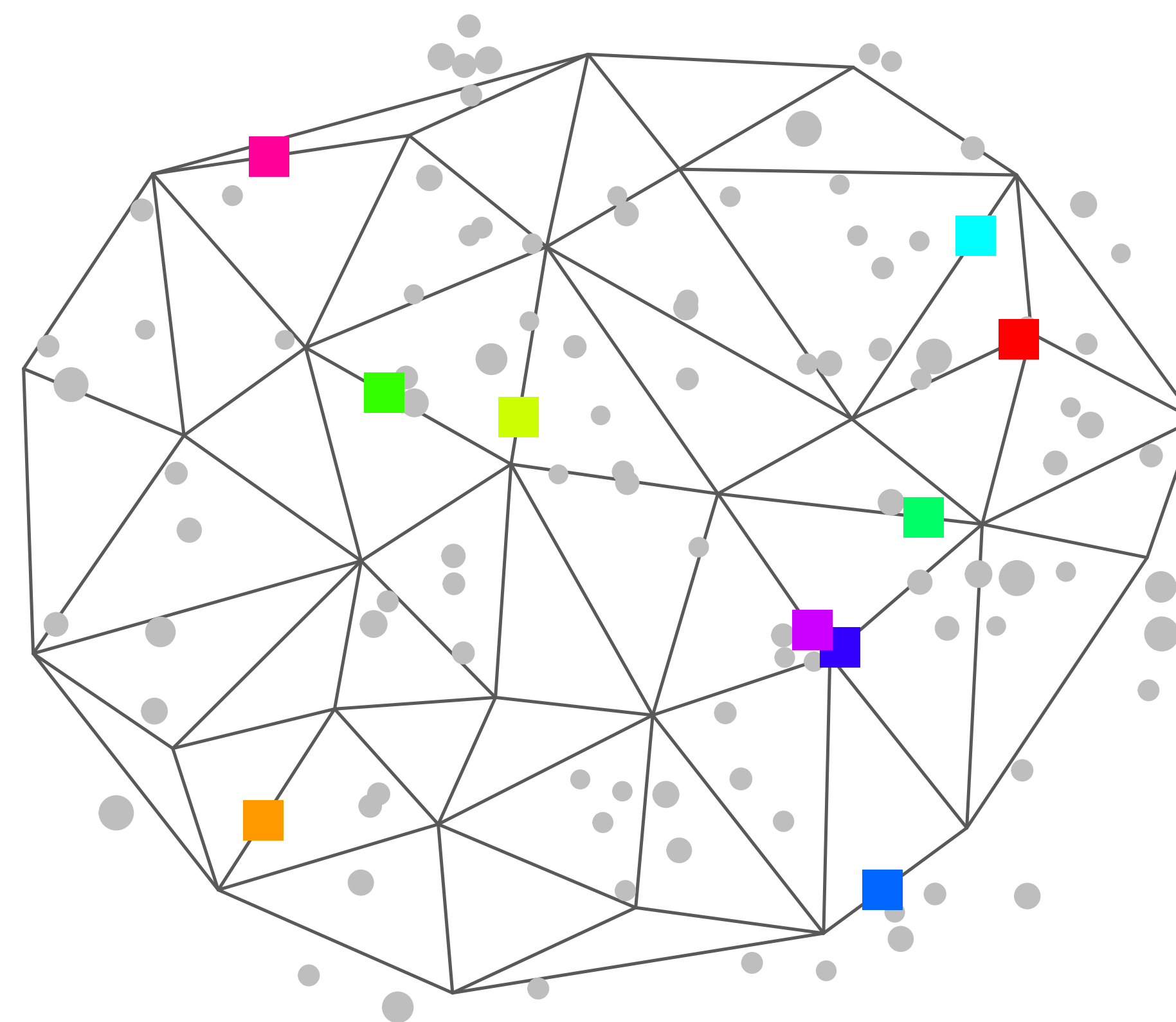
Straipsnio apžvalga: net-k-means algoritmas

- **Net-k-means algoritmas** skirtas rasti lokaliam klasteriavimo uždavinio su tinklo ribojimu centrams. **Šablonas:**
 1. Atsitiktinai parenkam pradiniai centrus $Q_1, Q_2, \dots, Q_K$ (kuriuos vėliau tikslinsim)



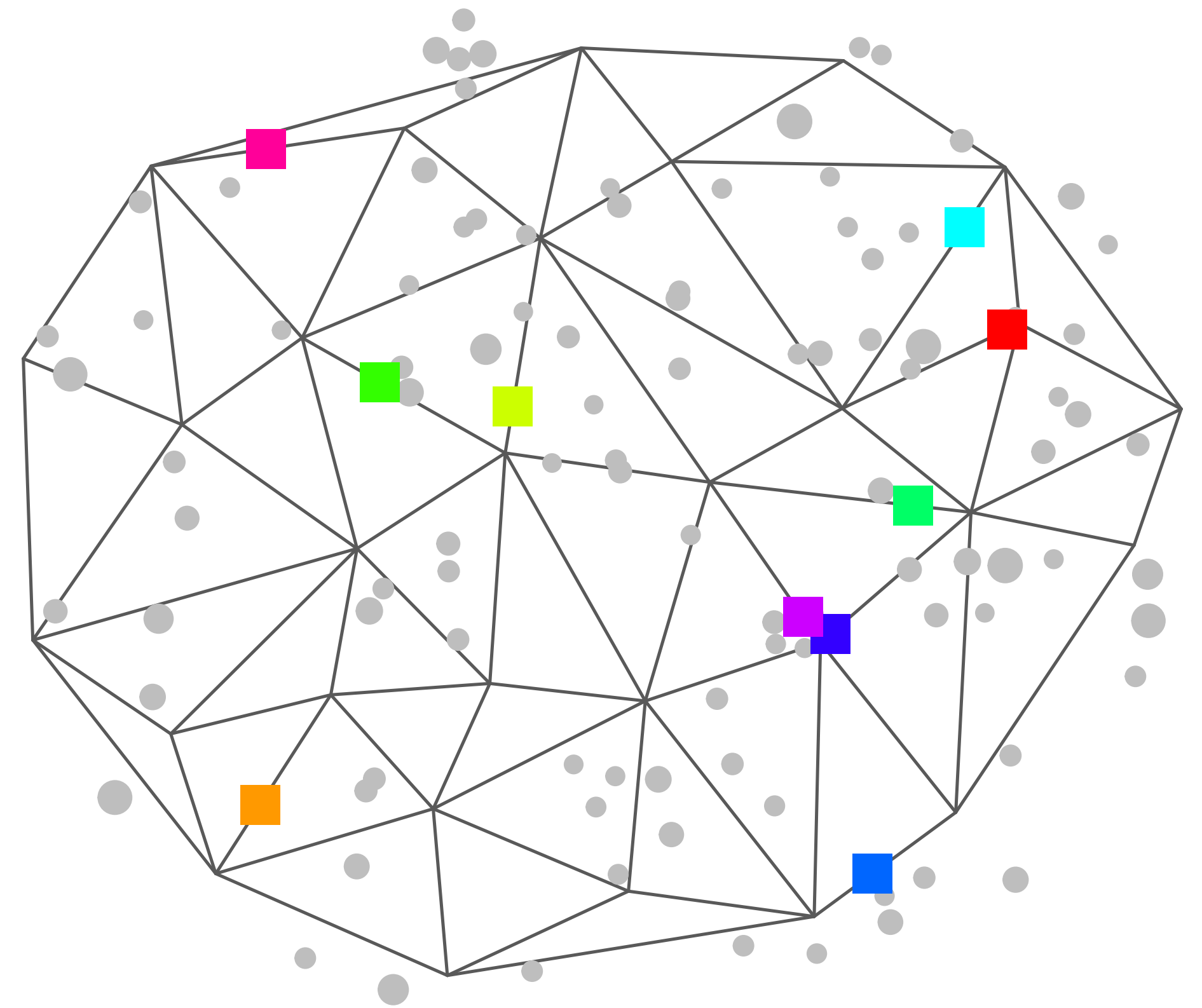
Straipsnio apžvalga: net-k-means algoritmas

- **Net-k-means algoritmas** skirtas rasti lokaliai klasteriavimo uždavinio su tinklo ribojimu centrams. **Šablonas:**
 1. Atsitiktinai parenkam pradiniai centrus $Q_1, Q_2, \dots, Q_K$ (kuriuos vėliau tikslinsim)



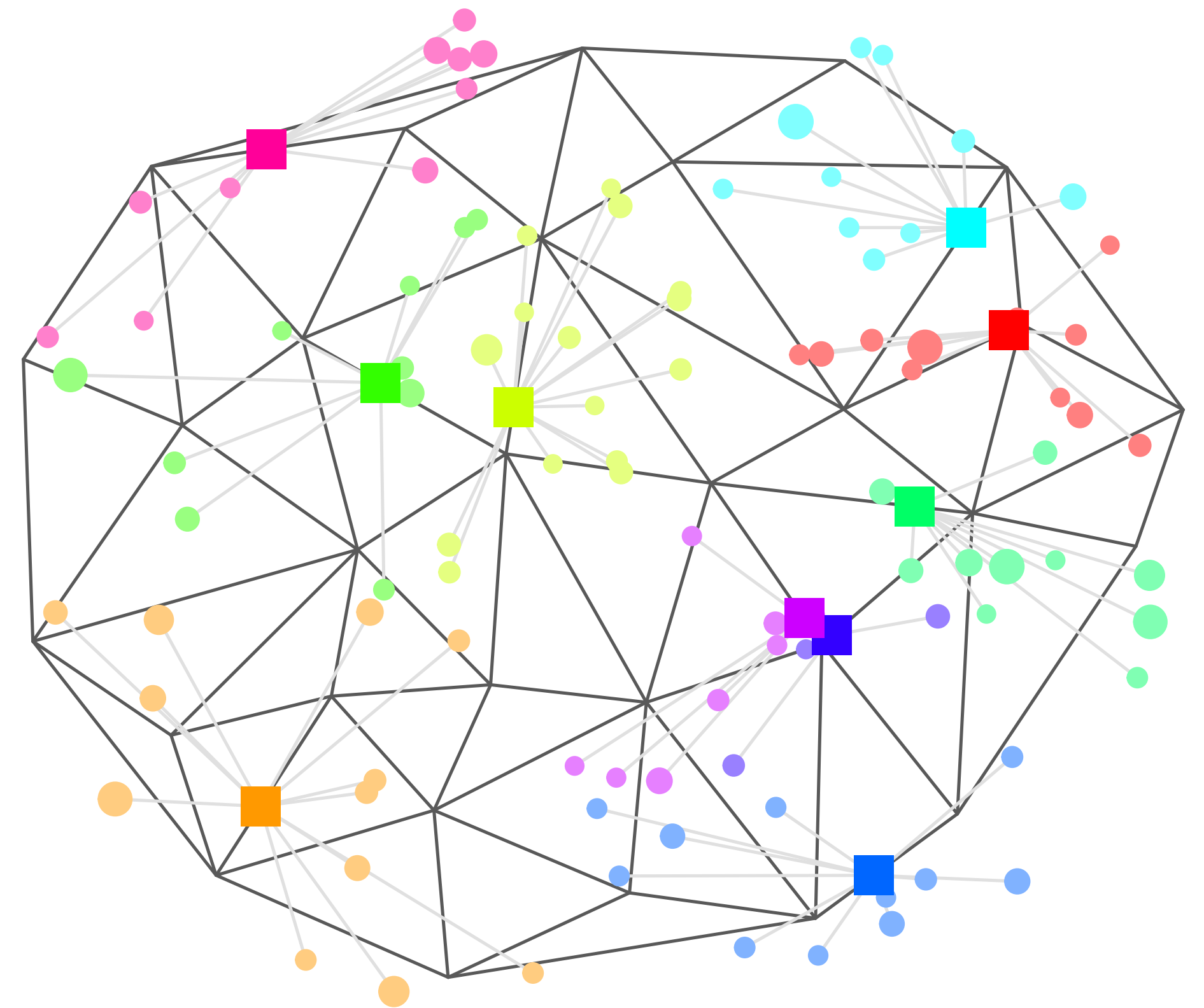
Straipsnio apžvalga: net-k-means algoritmas

- **Net-k-means algoritmas** skirtas rasti lokaliam klasteriavimo uždavinio su tinklo ribojimu centrams. **Šablonas:**
 1. Atsitiktinai parenkam pradiniai centrai $Q_1, Q_2, \dots, Q_K$ (kuriuos vėliau tikslinsim)
 2. **Priskyrimo Žingsnis.** Random klasterius $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_K$: kiekvienas taškas P_i yra priskiriamas artimiausiam centrui Q_k (ir atitinkamam klasteriui)



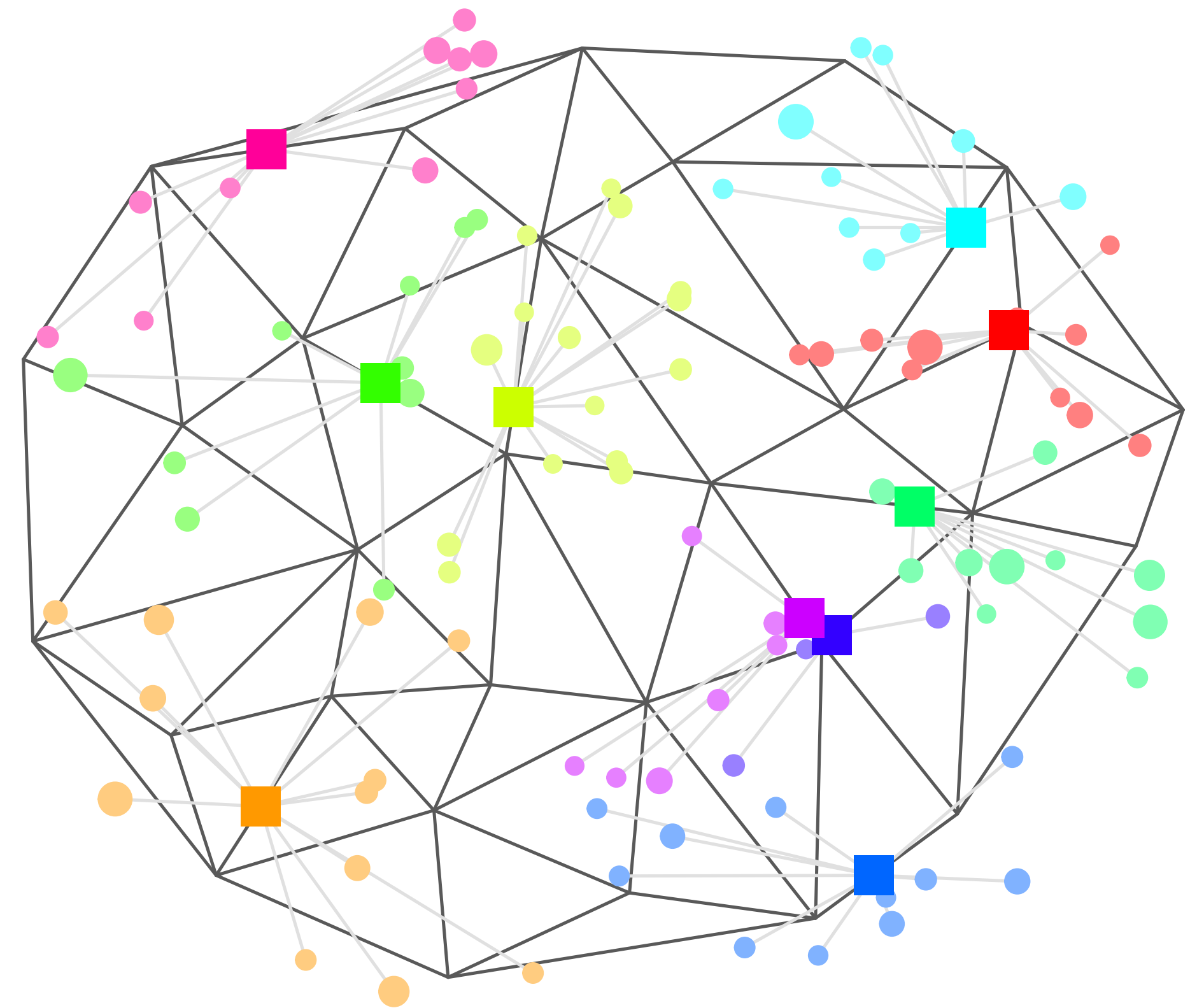
Straipsnio apžvalga: net-k-means algoritmas

- **Net-k-means algoritmas** skirtas rasti lokaliai klasteriavimo uždavinio su tinklo ribojimu centrums. **Šablonas:**
 1. Atsitiktinai parenkam pradiniai centrus $Q_1, Q_2, \dots, Q_K$ (kuriuos vėliau tikslinsim)
 2. **Priskyrimo Žingsnis.** Random klasterius $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_K$: kiekvienas taškas P_i yra priskiriamas artimiausiam centrui Q_k (ir atitinkamam klasteriui)



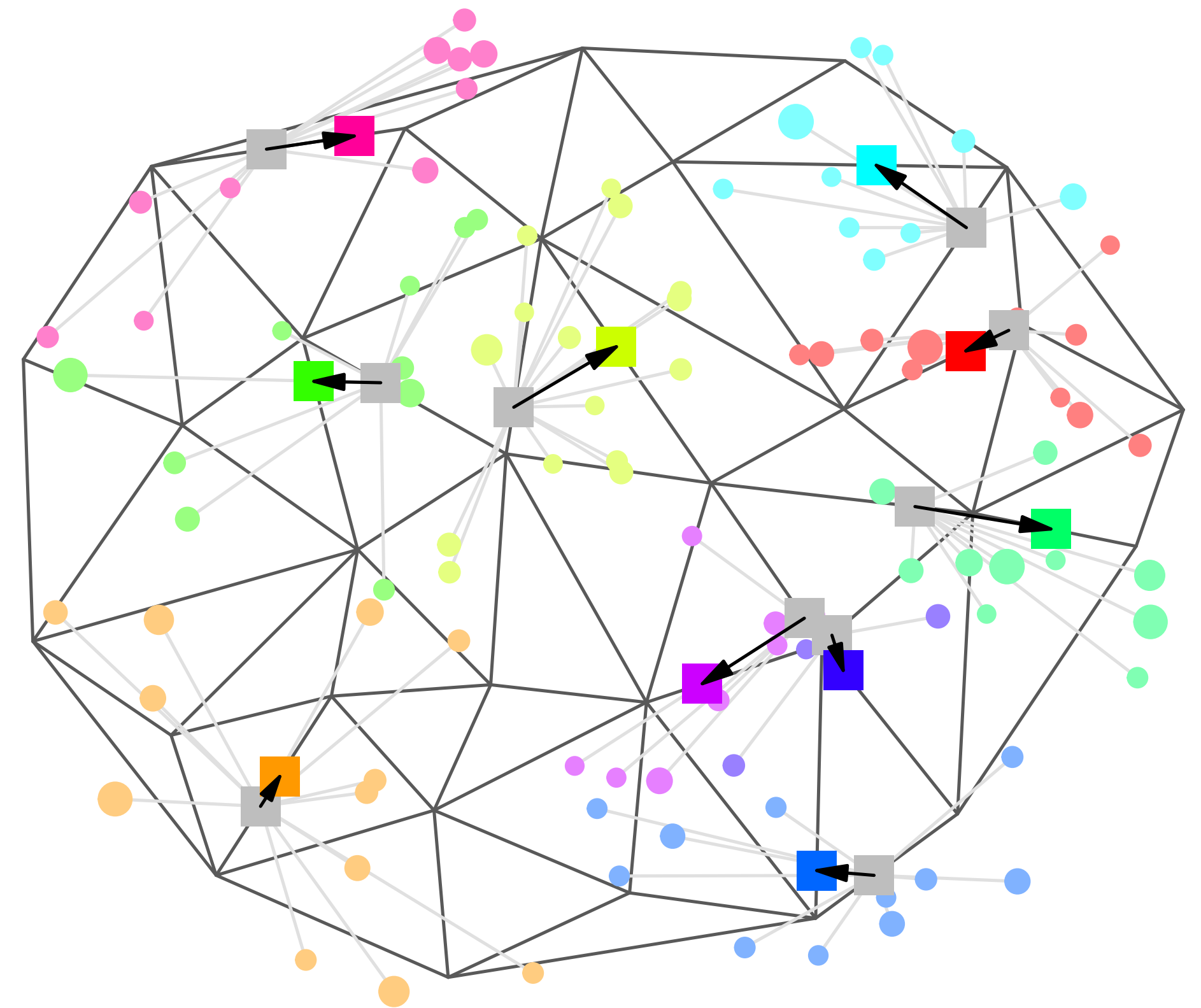
Straipsnio apžvalga: net-k-means algoritmas

- **Net-k-means algoritmas** skirtas rasti lokaliam klasteriavimo uždavinio su tinklo ribojimu centrums. **Šablonas:**
 1. Atsitiktinai parenkam pradinius centrus $Q_1, Q_2, \dots, Q_K$ (kuriuos vėliau tikslinsim)
 2. **Priskyrimo Žingsnis.** Randam klasterius $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_K$: kiekvienas taškas P_i yra priskiriamas artimiausiam centrui Q_k (ir atitinkamam klasteriui)
 3. **Centrų Tikslinimo Žingsnis.** Kiekvienam klasteriui $\mathcal{C}_k$ randame centrą Q_k kuris tenkina tinklo ribojimą $Q_k \in \mathcal{N}$ ir minimizuoja vidinius klasterio nuostolius



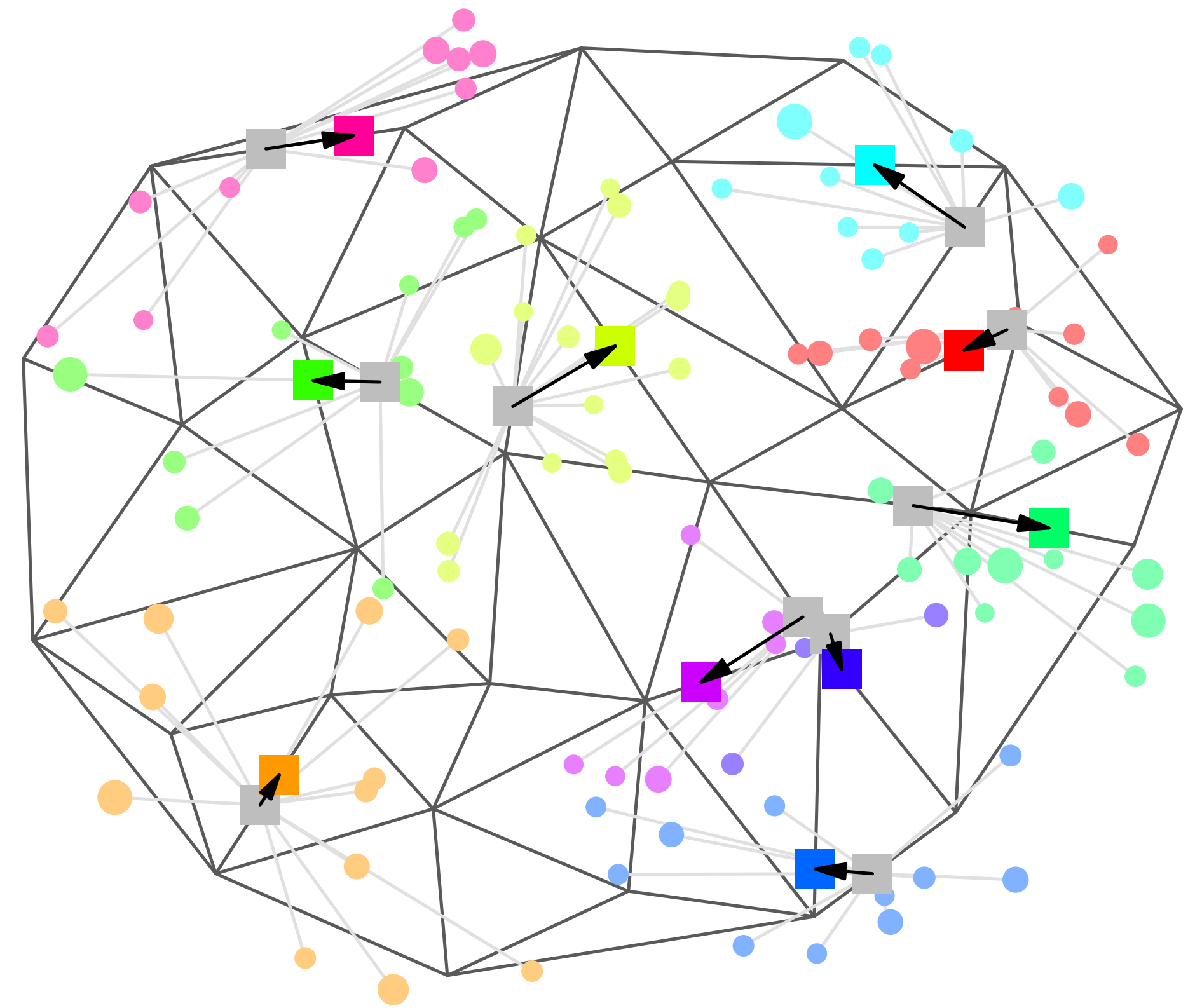
Straipsnio apžvalga: net-k-means algoritmas

- **Net-k-means algoritmas** skirtas rasti lokaliam klasteriavimo uždavinio su tinklo ribojimu centrums. **Šablonas:**
 1. Atsitiktinai parenkam pradinis centrus $Q_1, Q_2, \dots, Q_K$ (kuriuos vėliau tikslinsim)
 2. **Priskyrimo Žingsnis.** Random klasterius $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_K$: kiekvienas taškas P_i yra priskiriamas artimiausiam centrui Q_k (ir atitinkamam klasteriui)
 3. **Centrų Tikslinimo Žingsnis.** Kiekvienam klasteriui $\mathcal{C}_k$ randame centrą Q_k kuris tenkina tinklo ribojimą $Q_k \in \mathcal{N}$ ir minimizuoja vidinius klasterio nuostolius



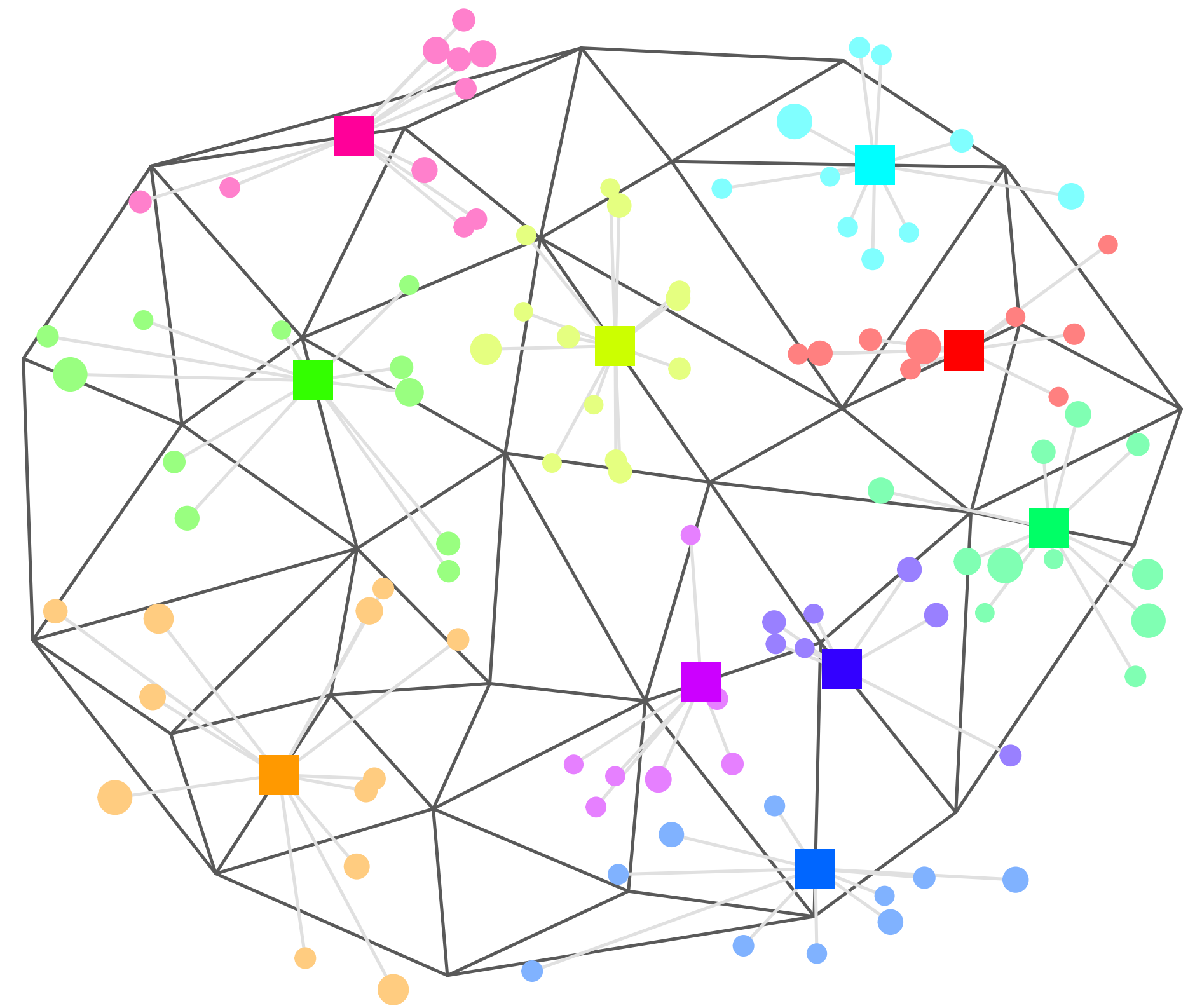
Straipsnio apžvalga: net-k-means algoritmas

- **Net-k-means algoritmas** skirtas rasti lokaliam klasteriavimo uždavinio su tinklo ribojimu centrums. **Šablonas:**
 1. Atsitiktinai parenkam pradinius centrus $Q_1, Q_2, \dots, Q_K$ (kuriuos vėliau tikslinsim)
 2. **Priskyrimo Žingsnis.** Random klasterius $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_K$: kiekvienas taškas P_i yra priskiriamas artimiausiam centrui Q_k (ir atitinkamam klasteriui)
 3. **Centrų Tikslinimo Žingsnis.** Kiekvienam klasteriui $\mathcal{C}_k$ randame centrą Q_k kuris tenkina tinklo ribojimą $Q_k \in \mathcal{N}$ ir minimizuoja vidinius klasterio nuostolius
 4. Kartojame **Priskyrimo ir Centrų Tikslinimo Žingsnius** kol klasteriai ir jų centrai stabilizuojasi



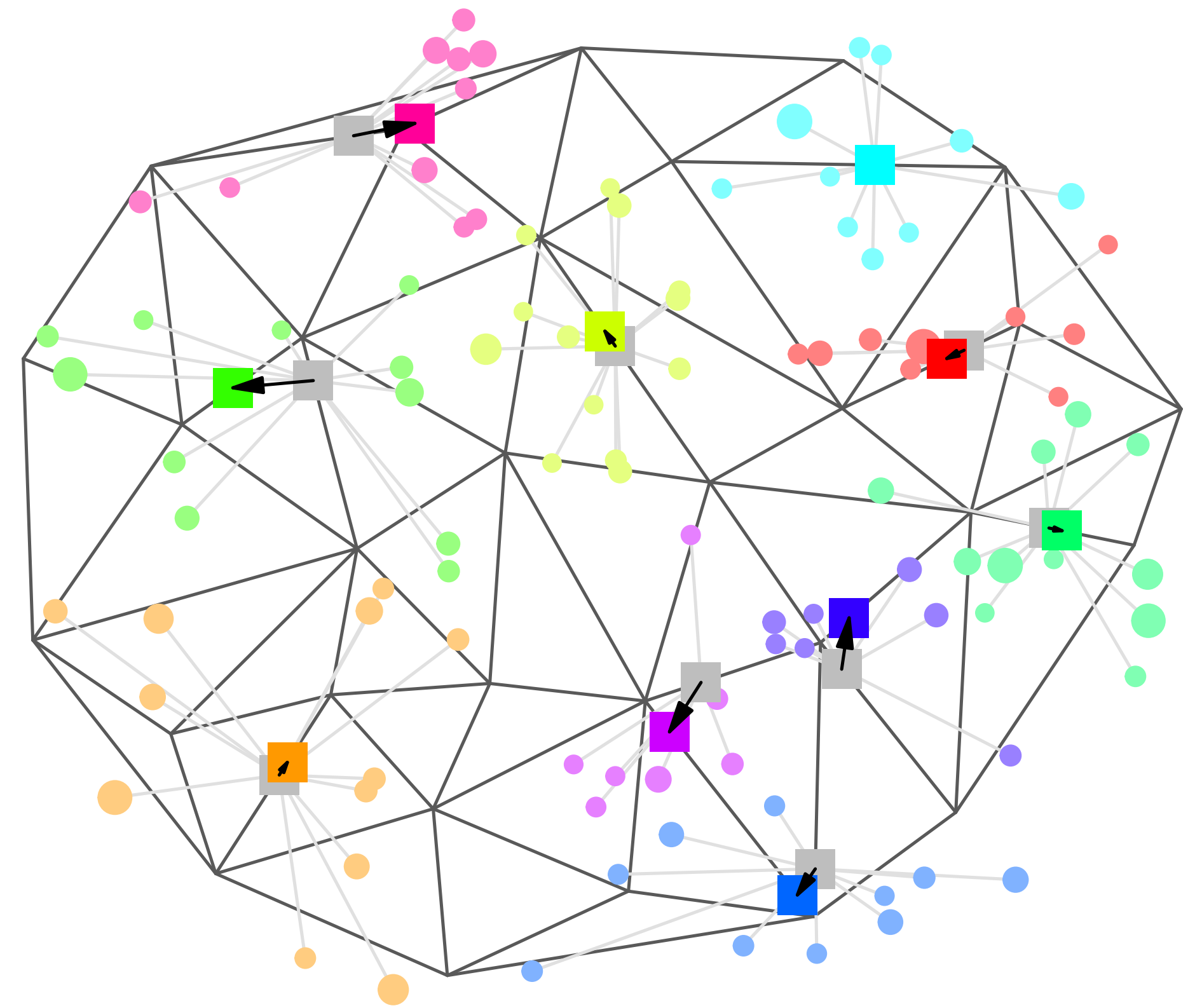
Straipsnio apžvalga: net-k-means algoritmas

- **Net-k-means algoritmas** skirtas rasti lokaliam klasteriavimo uždavinio su tinklo ribojimu centrums. **Šablonas:**
 1. Atsitiktinai parenkam pradinis centrus $Q_1, Q_2, \dots, Q_K$ (kuriuos vėliau tikslinsim)
 2. **Priskyrimo Žingsnis.** Random klasterius $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_K$: kiekvienas taškas P_i yra priskiriamas artimiausiam centrui Q_k (ir atitinkamam klasteriui)
 3. **Centrų Tikslinimo Žingsnis.** Kiekvienam klasteriui $\mathcal{C}_k$ randame centrą Q_k kuris tenkina tinklo ribojimą $Q_k \in \mathcal{N}$ ir minimizuoja vidinius klasterio nuostolius
 4. Kartojame **Priskyrimo ir Centrų Tikslinimo Žingsnius** kol klasteriai ir jų centrai stabilizuojasi



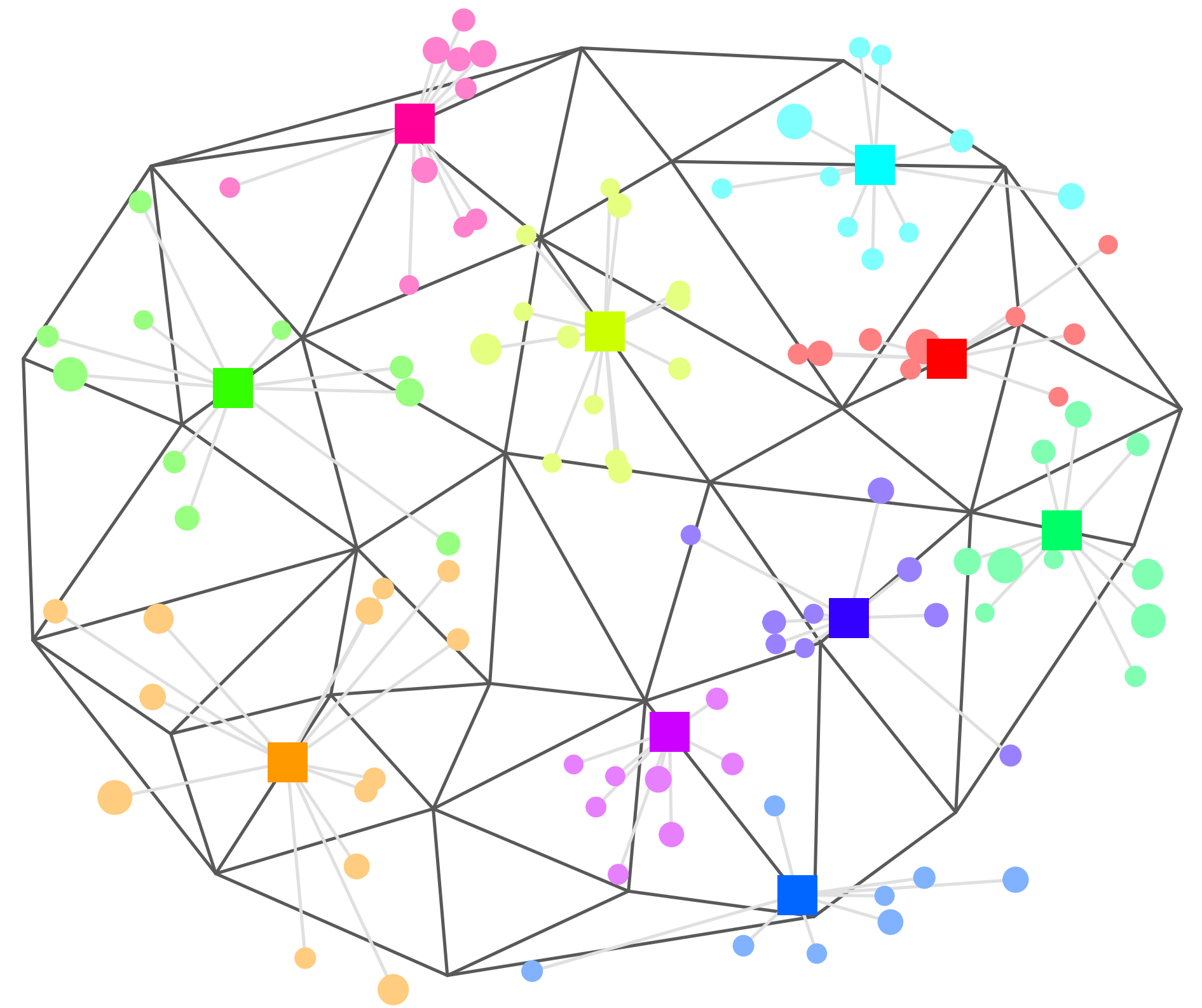
Straipsnio apžvalga: net-k-means algoritmas

- **Net-k-means algoritmas** skirtas rasti lokaliam klasteriavimo uždavinio su tinklo ribojimu centrums. **Šablonas:**
 1. Atsitiktinai parenkam pradinis centrus $Q_1, Q_2, \dots, Q_K$ (kuriuos vėliau tikslinsim)
 2. **Priskyrimo Žingsnis.** Random klasterius $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_K$: kiekvienas taškas P_i yra priskiriamas artimiausiam centrui Q_k (ir atitinkamam klasteriui)
 3. **Centrų Tikslinimo Žingsnis.** Kiekvienam klasteriui $\mathcal{C}_k$ randame centrą Q_k kuris tenkina tinklo ribojimą $Q_k \in \mathcal{N}$ ir minimizuoja vidinius klasterio nuostolius
 4. Kartojame **Priskyrimo ir Centrų Tikslinimo Žingsnius** kol klasteriai ir jų centrai stabilizuojasi



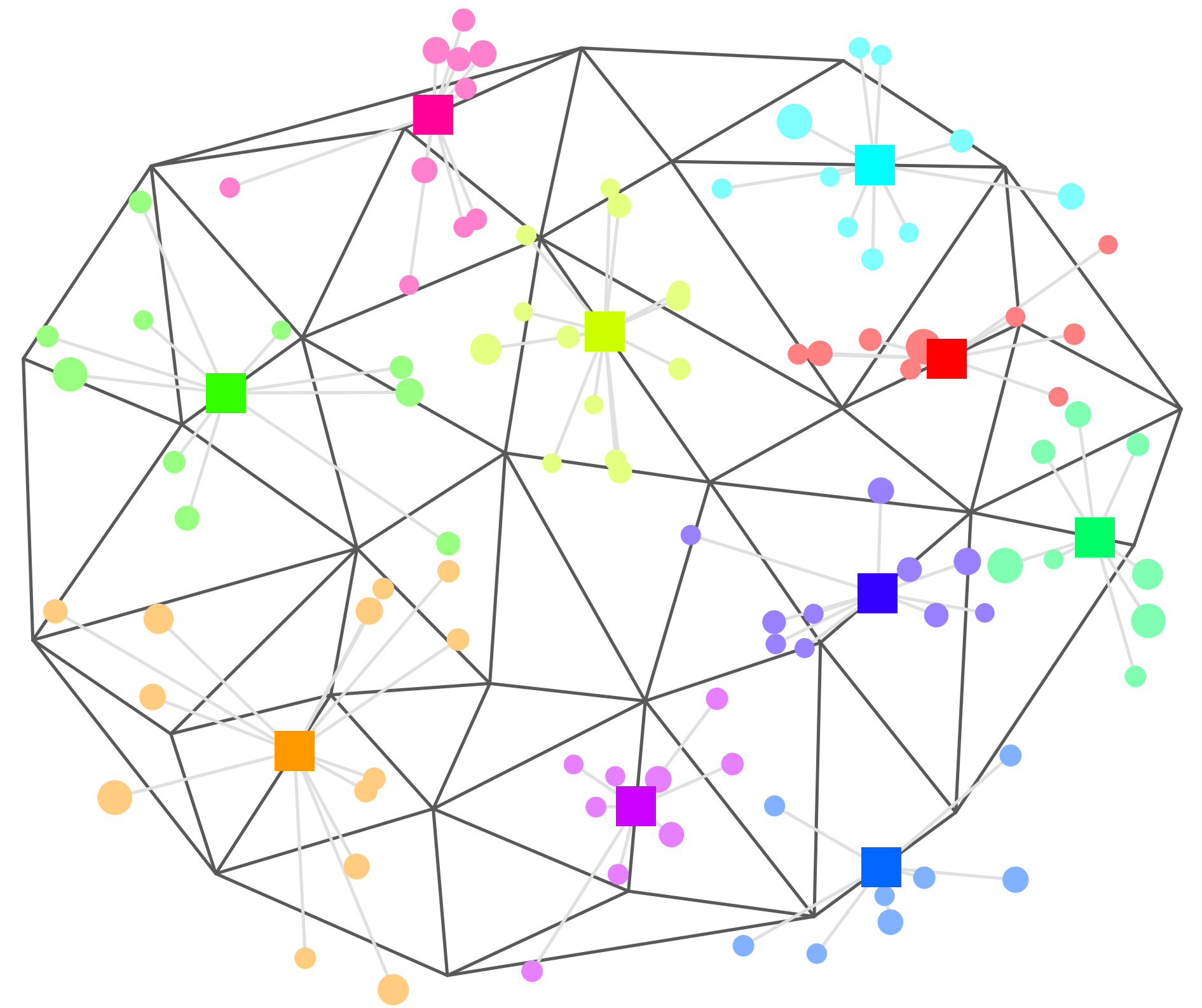
Straipsnio apžvalga: net-k-means algoritmas

- **Net-k-means algoritmas** skirtas rasti lokaliam klasteriavimo uždavinio su tinklo ribojimu centrums. **Šablonas:**
 1. Atsitiktinai parenkam pradinius centrus $Q_1, Q_2, \dots, Q_K$ (kuriuos vėliau tikslinsim)
 2. **Priskyrimo Žingsnis.** Random klasterius $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_K$: kiekvienas taškas P_i yra priskiriamas artimiausiam centrui Q_k (ir atitinkamam klasteriui)
 3. **Centrų Tikslinimo Žingsnis.** Kiekvienam klasteriui $\mathcal{C}_k$ randame centrą Q_k kuris tenkina tinklo ribojimą $Q_k \in \mathcal{N}$ ir minimizuoja vidinius klasterio nuostolius
 4. Kartojame **Priskyrimo ir Centrų Tikslinimo Žingsnius** kol klasteriai ir jų centrai stabilizuojasi



Straipsnio apžvalga: net-k-means algoritmas

- **Net-k-means algoritmas** skirtas rasti lokaliam klasteriavimo uždavinio su tinklo ribojimu centrums. **Šablonas:**
 1. Atsitiktinai parenkam pradinis centrus $Q_1, Q_2, \dots, Q_K$ (kuriuos vėliau tikslinsim)
 2. **Priskyrimo Žingsnis.** Random klasterius $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_K$: kiekvienas taškas P_i yra priskiriamas artimiausiam centrui Q_k (ir atitinkamam klasteriui)
 3. **Centrų Tikslinimo Žingsnis.** Kiekvienam klasteriui $\mathcal{C}_k$ randame centrą Q_k kuris tenkina tinklo ribojimą $Q_k \in \mathcal{N}$ ir minimizuoja vidinius klasterio nuostolius
 4. Kartojame **Priskyrimo ir Centrų Tikslinimo Žingsnius** kol klasteriai ir jų centrai stabilizuojasi



Straipsnio apžvalga: globalusis sprendinys

Straipsnio apžvalga: globalusis sprendinys

- **Uždavinį galima suformuluoti kaip mixed-integer-quadratically-constrained-programming (MIQCP) uždavinį**

Straipsnio apžvalga: globalusis sprendinys

- Uždavinį galima suformuluoti kaip **mixed-integer-quadratically-constrained-programming (MIQCP)** uždavinį
- Suformulavom uždavinį “gurobi” kalba

Straipsnio apžvalga: globalusis sprendinys

- Uždavinį galima suformuluoti kaip **mixed-integer-quadratically-constrained-programming (MIQCP)** uždavinį
- Suformulavom uždavinį “gurobi” kalba
- Paleidom “kompiuterinį” eksperimentą:

Straipsnio apžvalga: globalusis sprendinys

- Uždavinį galima suformuluoti kaip **mixed-integer-quadratically-constrained-programming (MIQCP)** uždavinį
- Suformulavom uždavinį “gurobi” kalba
- Paleidom “kompiuterinį” eksperimentą:

for $N \in \{10, 15, \dots, 45, 50\}$ # taškų skaičius

Straipsnio apžvalga: globalusis sprendinys

- Uždavinį galima suformuluoti kaip **mixed-integer-quadratically-constrained-programming (MIQCP)** uždavinį
- Suformulavom uždavinį “gurobi” kalba
- Paleidom “kompiuterinį” eksperimentą:

for $N \in \{10, 15, \dots, 45, 50\}$ # taškų skaičius

for $M \in \{20, 30, \dots, 90, 100\}$ # tinklo briaunų skaičius

Straipsnio apžvalga: globalusis sprendinys

- Uždavinį galima suformuluoti kaip **mixed-integer-quadratically-constrained-programming (MIQCP)** uždavinį
- Suformulavom uždavinį “gurobi” kalba
- Paleidom “kompiuterinį” eksperimentą:

for $N \in \{10, 15, \dots, 45, 50\}$ # taškų skaičius

for $M \in \{20, 30, \dots, 90, 100\}$ # tinklo briaunų skaičius

for $K \in \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$ # klasterių skaičius

Straipsnio apžvalga: globalusis sprendinys

- Uždavinį galima suformuluoti kaip **mixed-integer-quadratically-constrained-programming (MIQCP)** uždavinį
- Suformulavom uždavinį “gurobi” kalba
- Paleidom “kompiuterinį” eksperimentą:

```
for  $N \in \{10, 15, \dots, 45, 50\}$       # taškų skaičius  
    for  $M \in \{20, 30, \dots, 90, 100\}$   # tinklo briaunų skaičius  
        for  $K \in \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$   # klasterių skaičius  
            solve_MIQCP_program_with_gurobi( $N, M, K$ )
```

Straipsnio apžvalga: globalusis sprendinys

- Uždavinį galima suformuluoti kaip **mixed-integer-quadratically-constrained-programming (MIQCP)** uždavinį
- Suformulavom uždavinį “gurobi” kalba
- Paleidom “kompiuterinį” eksperimentą:

```
for  $N \in \{10, 15, \dots, 45, 50\}$       # taškų skaičius  
    for  $M \in \{20, 30, \dots, 90, 100\}$   # tinklo briaunų skaičius  
        for  $K \in \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$   # klasterių skaičius  
            solve_MIQCP_program_with_gurobi( $N, M, K$ )
```

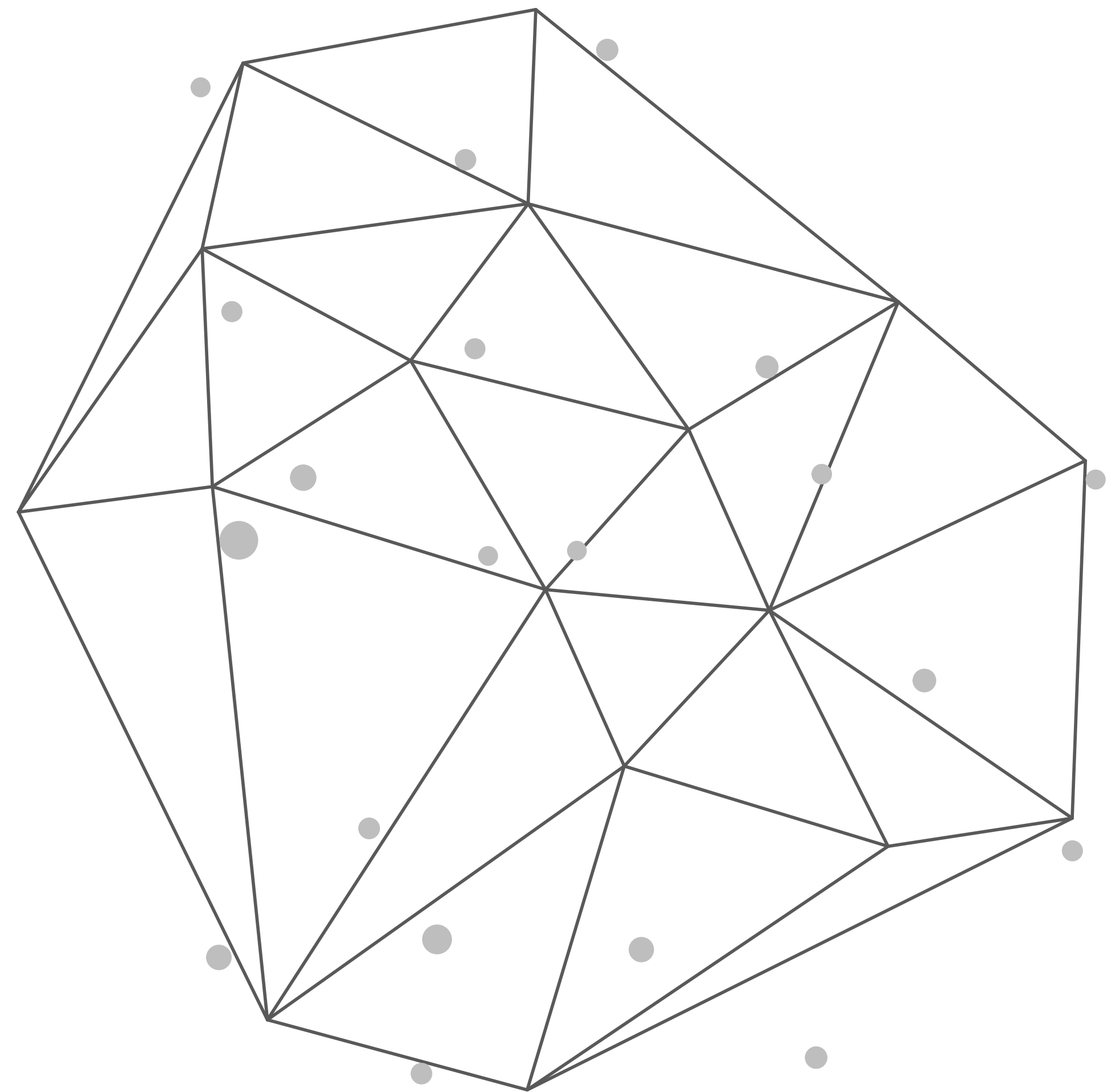
$\implies$ Rezultatai nuvylė: “gurobis” sugebėjo rasti globalų sprendinį tik labai “mažiems” uždavinio atvejams

Straipsnio apžvalga: globalusis sprendinys

- Uždavinį galima suformuluoti kaip **mixed-integer-quadratically-constrained-programming (MIQCP)** uždavinį
- Suformulavom uždavinį “gurobi” kalba
- Paleidom “kompiuterinį” eksperimentą:

```
for  $N \in \{10, 15, \dots, 45, 50\}$       # taškų skaičius  
  for  $M \in \{20, 30, \dots, 90, 100\}$   # tinklo briaunų skaičius  
    for  $K \in \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$   # klasterių skaičius  
      solve_MIQCP_program_with_gurobi( $N, M, K$ )
```

⇒ Rezultatai nuvylė: “gurobis” sugebėjo rasti globalų sprendinį tik labai “mažiems” uždavinio atvejams



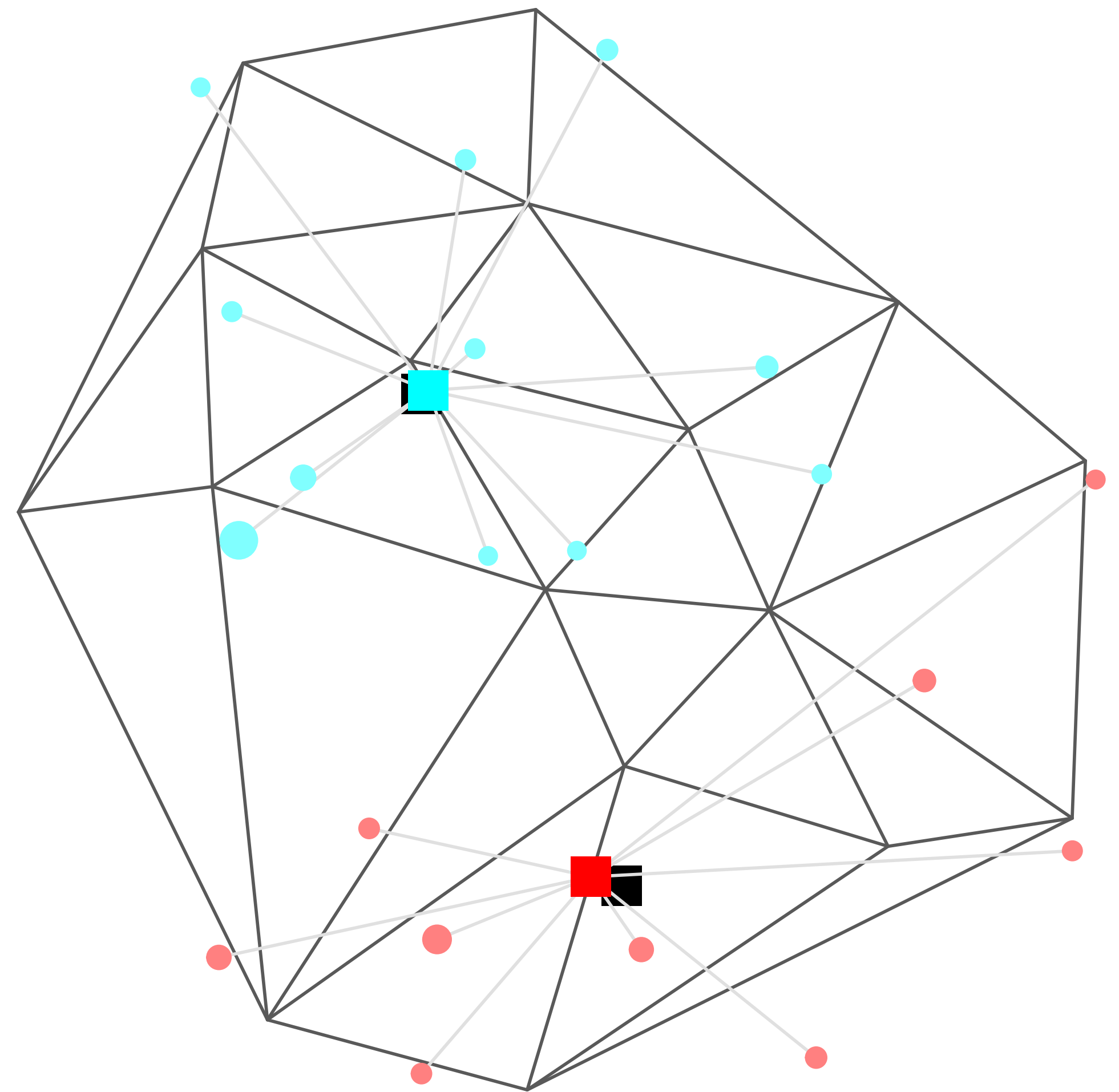
Straipsnio apžvalga: globalusis sprendinys

- Uždavinį galima suformuluoti kaip **mixed-integer-quadratically-constrained-programming (MIQCP)** uždavinį
- Suformulavom uždavinį “gurobi” kalba
- Paleidom “kompiuterinį” eksperimentą:

```
for  $N \in \{10, 15, \dots, 45, 50\}$       # taškų skaičius  
  for  $M \in \{20, 30, \dots, 90, 100\}$   # tinklo briaunų skaičius  
    for  $K \in \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$   # klasterių skaičius
```

```
      solve_MIQCP_program_with_gurobi( $N, M, K$ )
```

⇒ Rezultatai nuvylė: “gurobis” sugebėjo rasti globalų sprendinį tik labai “mažiems” uždavinio atvejams

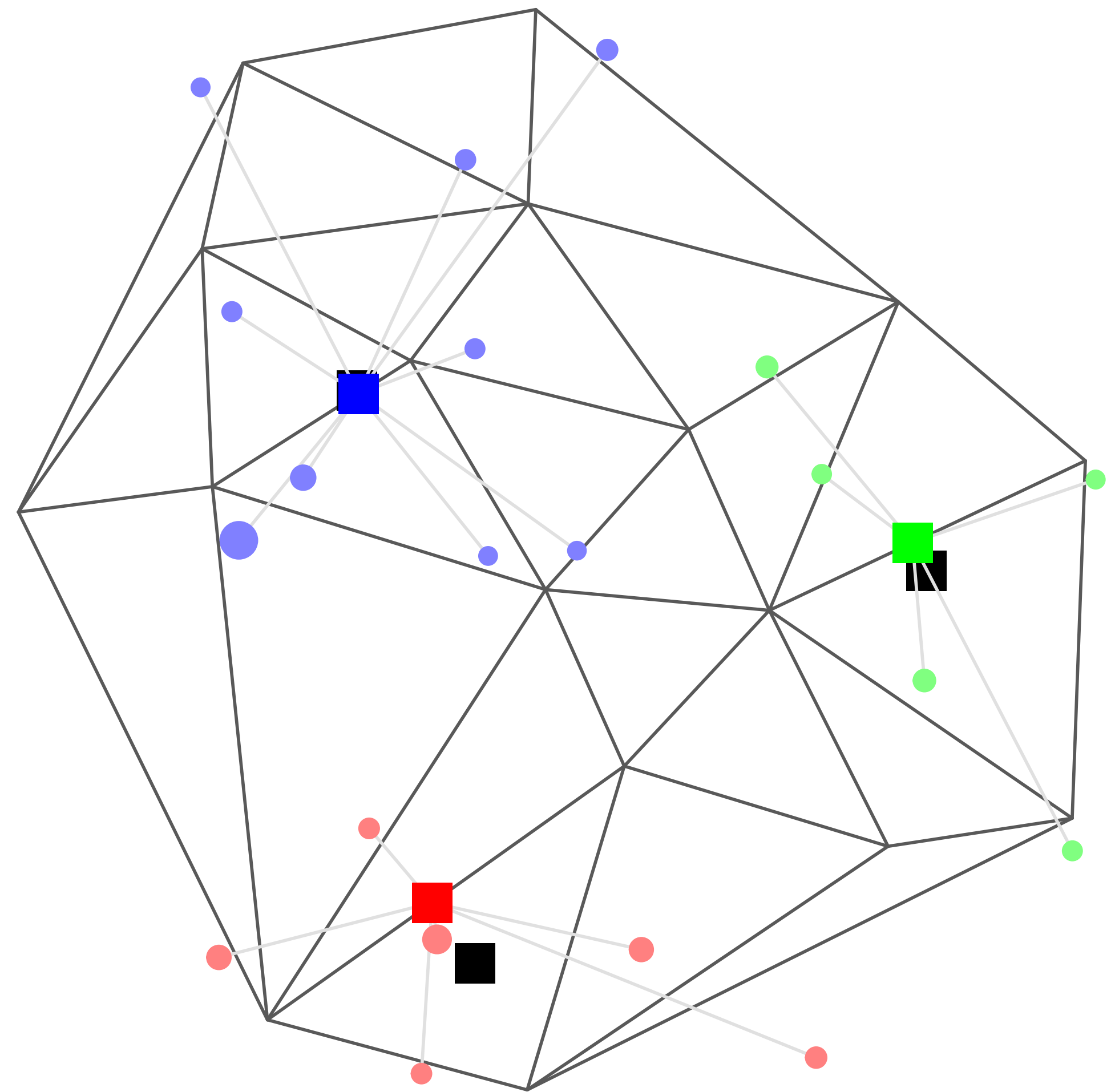


Straipsnio apžvalga: globalusis sprendinys

- Uždavinį galima suformuluoti kaip **mixed-integer-quadratically-constrained-programming (MIQCP)** uždavinį
- Suformulavom uždavinį “gurobi” kalba
- Paleidom “kompiuterinį” eksperimentą:

```
for  $N \in \{10, 15, \dots, 45, 50\}$       # taškų skaičius  
  for  $M \in \{20, 30, \dots, 90, 100\}$   # tinklo briaunų skaičius  
    for  $K \in \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$   # klasterių skaičius  
      solve_MIQCP_program_with_gurobi( $N, M, K$ )
```

⇒ Rezultatai nuvylė: “gurobis” sugebėjo rasti globalų sprendinį tik labai “mažiems” uždavinio atvejams

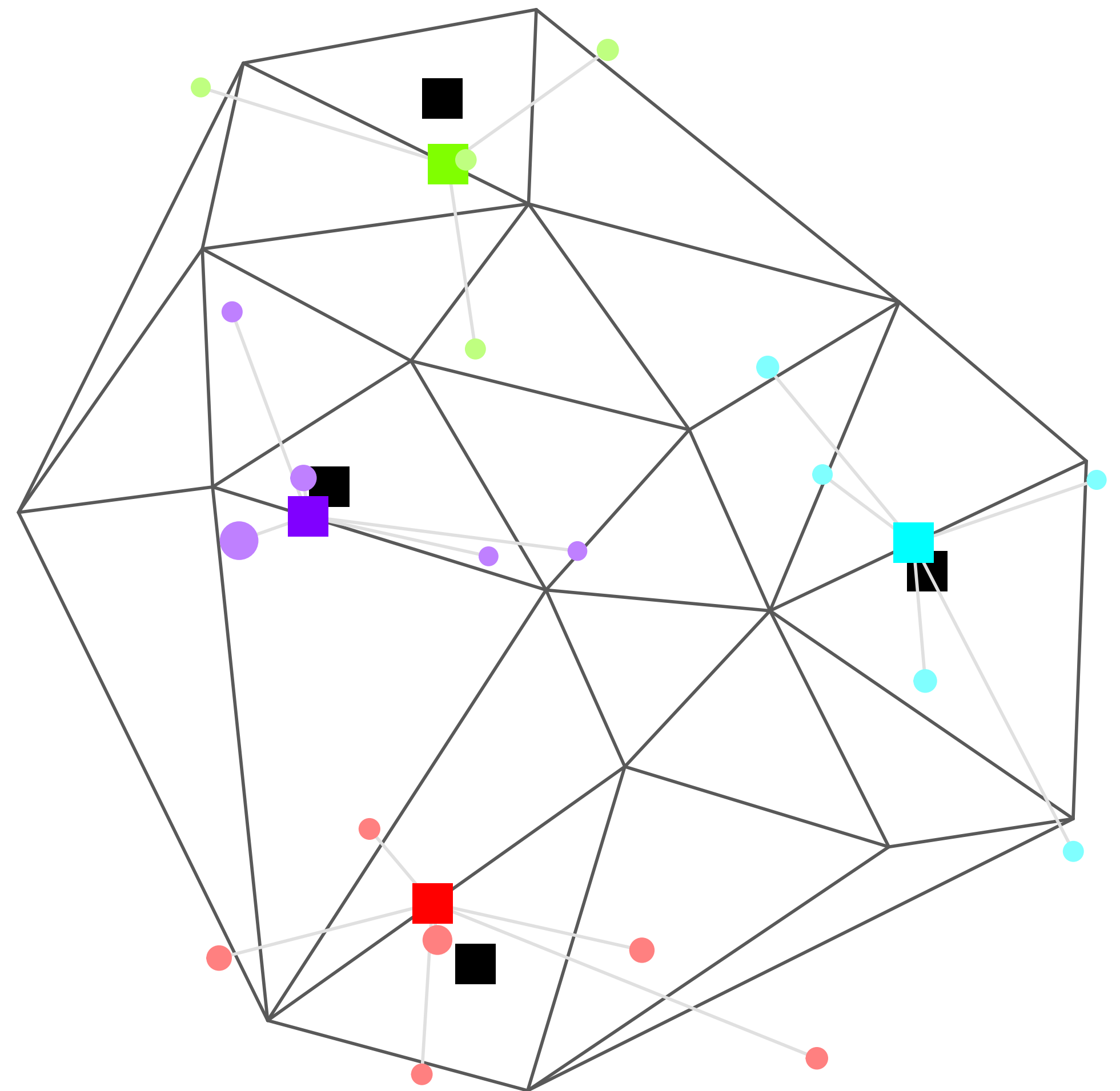


Straipsnio apžvalga: globalusis sprendinys

- Uždavinį galima suformuluoti kaip **mixed-integer-quadratically-constrained-programming (MIQCP)** uždavinį
- Suformulavom uždavinį “gurobi” kalba
- Paleidom “kompiuterinį” eksperimentą:

```
for  $N \in \{10, 15, \dots, 45, 50\}$       # taškų skaičius  
  for  $M \in \{20, 30, \dots, 90, 100\}$   # tinklo briaunų skaičius  
    for  $K \in \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$   # klasterių skaičius  
      solve_MIQCP_program_with_gurobi( $N, M, K$ )
```

⇒ Rezultatai nuvylė: “gurobis” sugebėjo rasti globalų sprendinį tik labai “mažiems” uždavinio atvejams



Straipsnio apžvalga: perrankos algoritmas

Straipsnio apžvalga: perrankos algoritmas

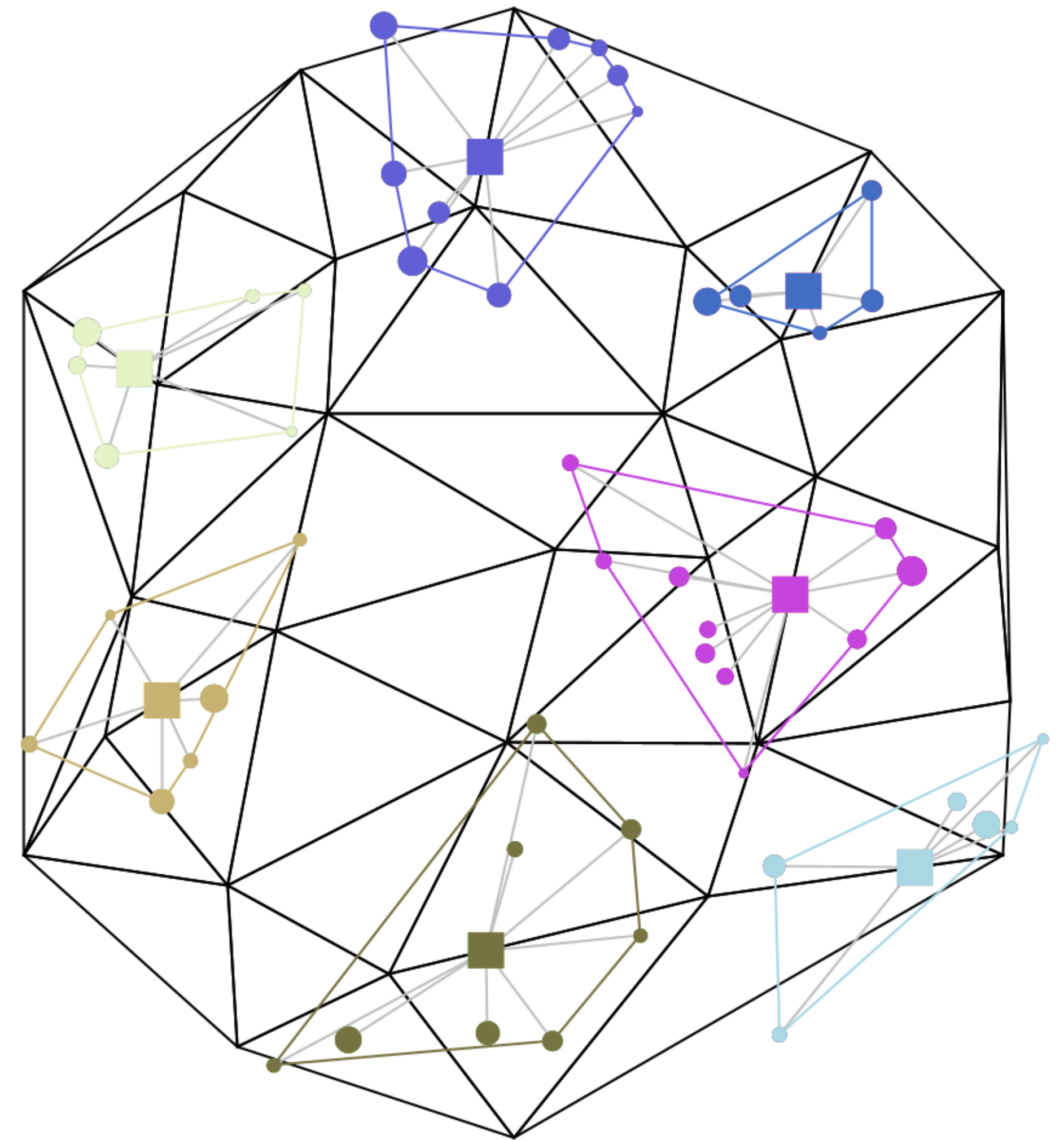
- Kadangi “gurobio” rezultatai nuvylė, pabandėme išspręsti “didesnius” uždavinius naudojant “protingą” perrankos algoritmą.

Straipsnio apžvalga: perrankos algoritmas

- Kadangi “gurobio” rezultatai nuvylė, pabandėme išspręsti “didesnius” uždavinius naudojant “protingą” perrankos algoritmą.
 $\implies$ Pavyko išspręsti uždavinį su 7 klasteriais ir 50 taškų.
Perrankos algoritmo laikas: ≈ 680 s. Laiko modelis prognozuoja, kad “gurobis” šiam uždaviniui būtų “užtrukęs” daugiau nei 1000 metų...

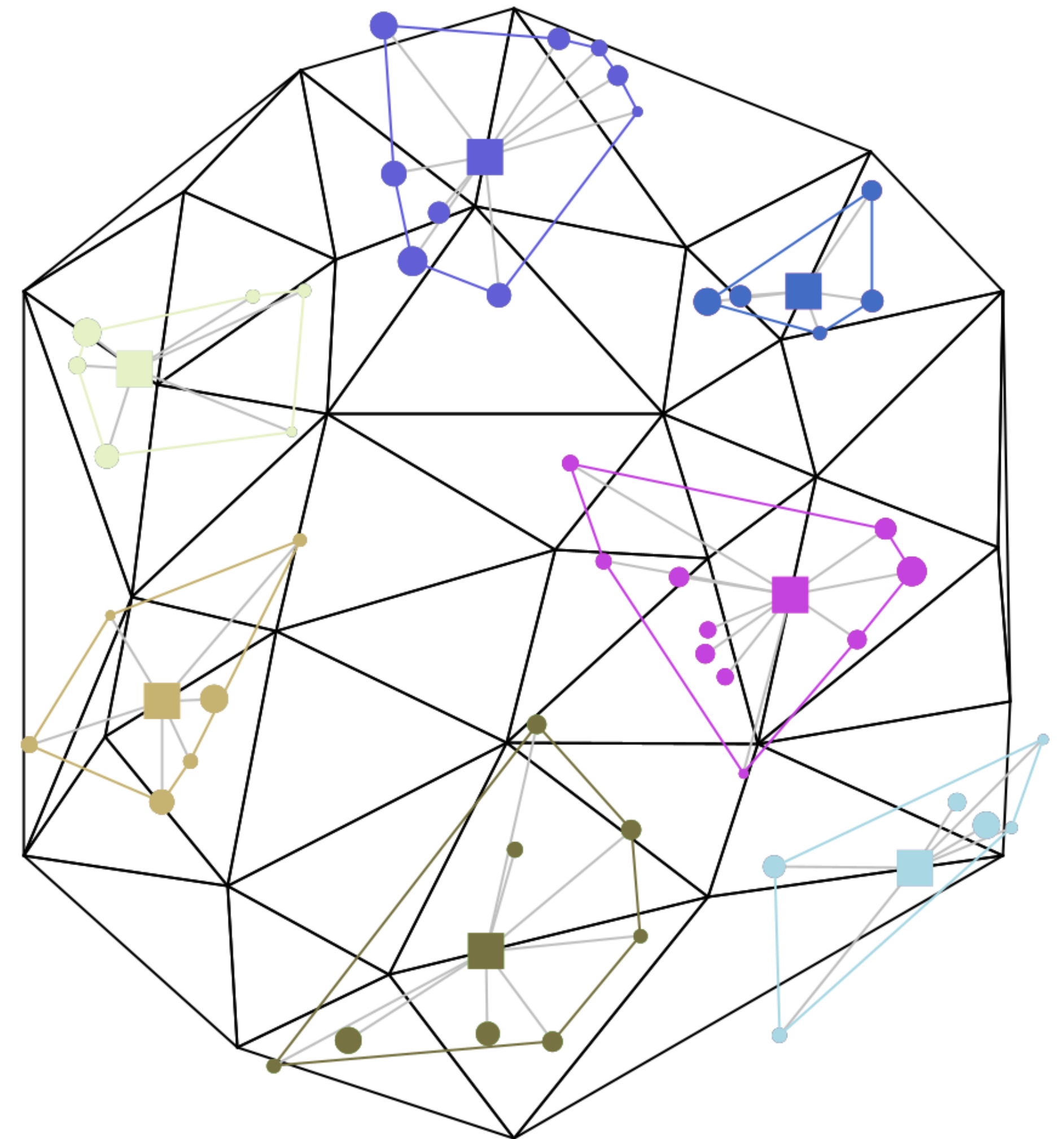
Straipsnio apžvalga: perrankos algoritmas

- Kadangi “gurobio” rezultatai nuvylė, pabandėme išspręsti “didesnius” uždavinius naudojant “protingą” perrankos algoritmą.
⇒ Pavyko išspręsti uždavinį su 7 klasteriais ir 50 taškų.
Perrankos algoritmo laikas: ≈ 680 s. Laiko modelis prognozuoja, kad “gurobis” šiam uždaviniui būtų “užtrukęs” daugiau nei 1000 metų...



Straipsnio apžvalga: perrankos algoritmas

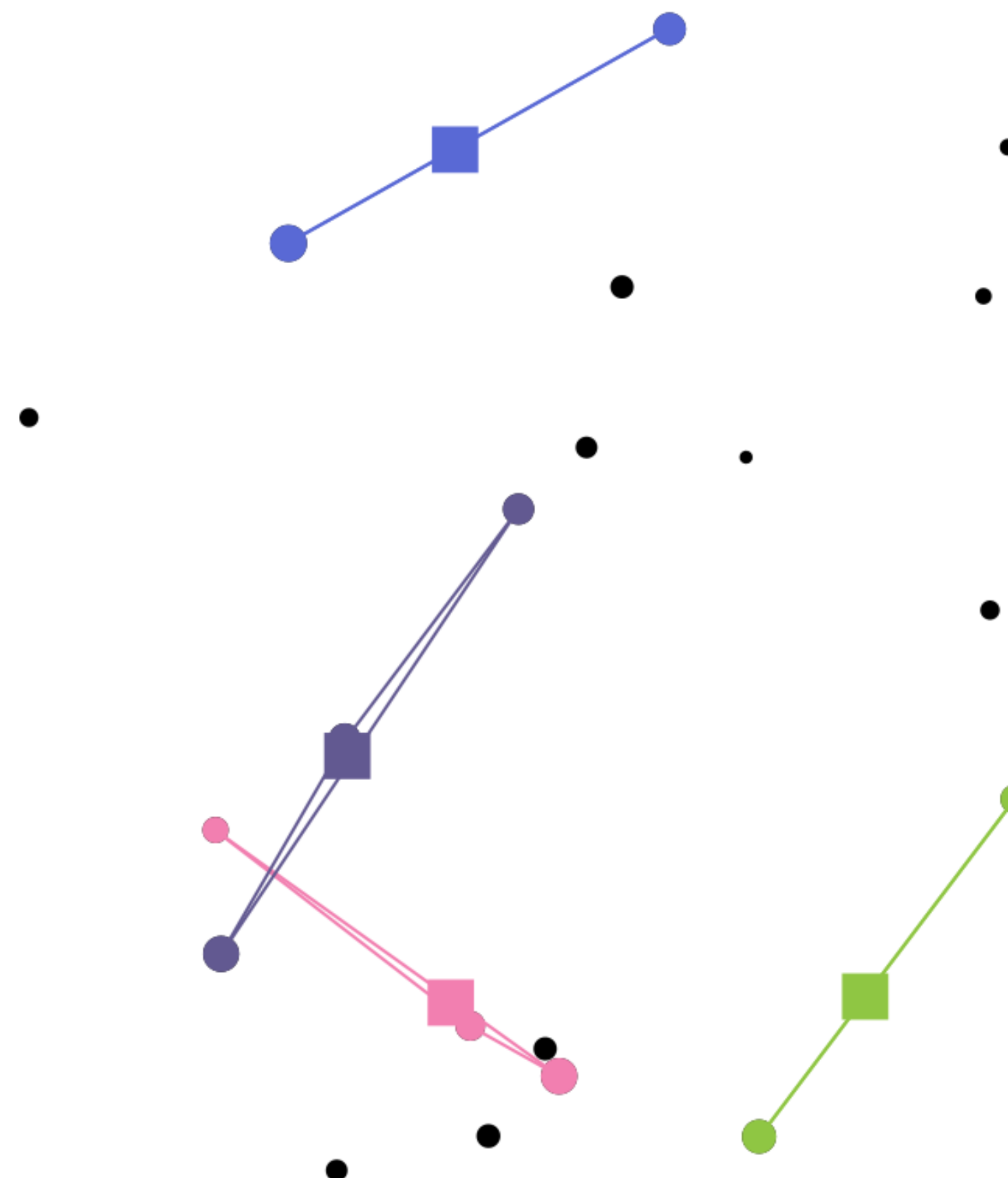
- Kadangi “gurobio” rezultatai nuvylė, pabandėme išspręsti “didesnius” uždavinius naudojant “protingą” perrankos algoritmą.
 - ⇒ Pavyko išspręsti uždavinį su 7 klasteriais ir 50 taškų. Perrankos algoritmo laikas: ≈ 680 s. Laiko modelis prognozuoja, kad “gurobis” šiam uždaviniui būtų “užtrukęs” daugiau nei 1000 metų...
 - ⇒ Vis dėlto lyginant su neapribotu klasteriavimo uždaviniu, šis rezultatas nėra įspūdingas: neseniai išėjęs straipsnis aprašo būdą, kaip išspręsti uždavinį su daugiau nei 1000 taškų, kai klasterių skaičius yra palyginti nedidelis (≤ 10). (Tačiau šis metodas negali būti tiesiogiai pritaikytas uždaviniams su ribojimais klasterių centrams $Q \in \mathcal{N}$.)



Perrankos algoritmas: šakų “pjovimo” kriterijus

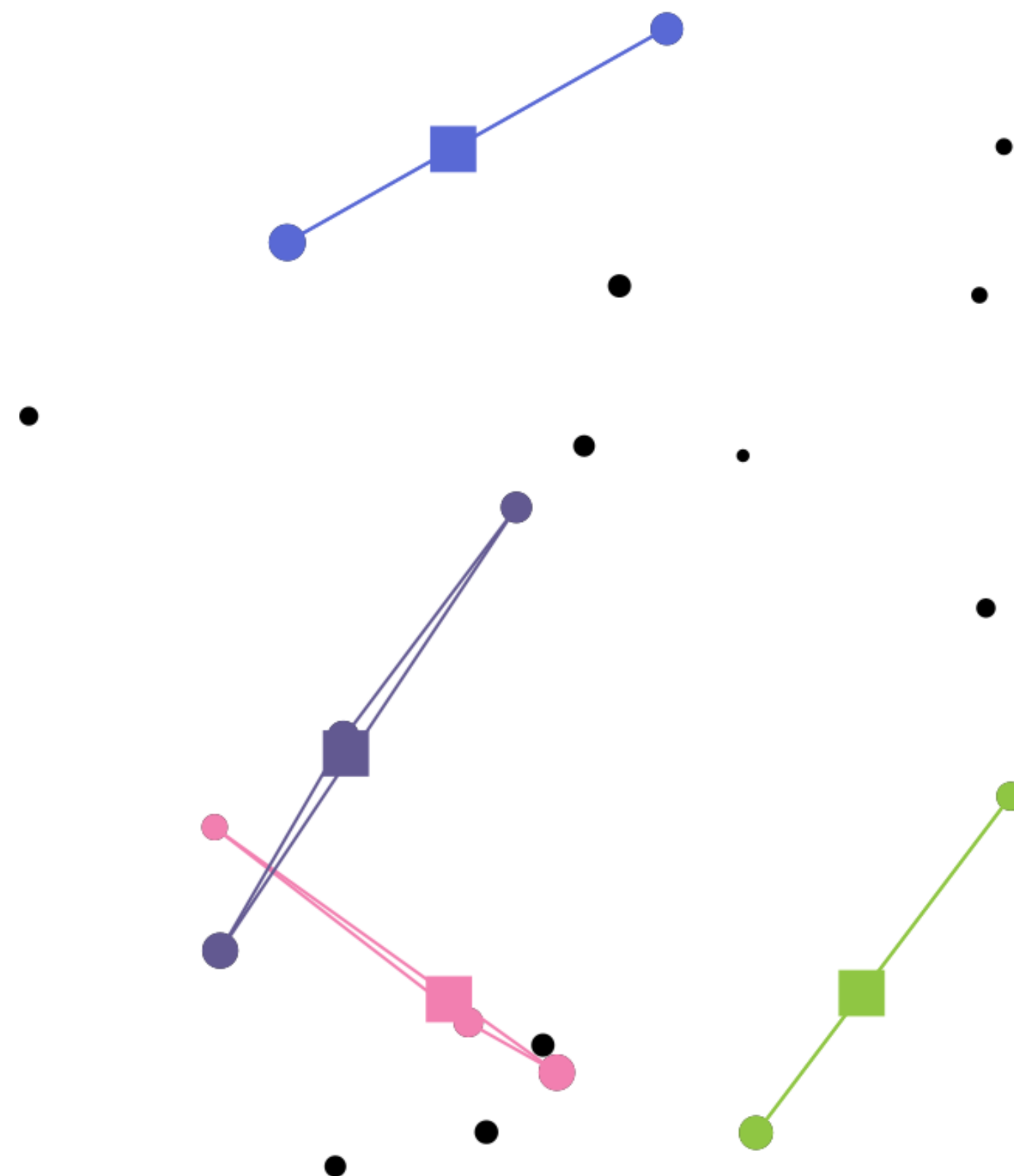
Perrankos algoritmas: šakų “pjovimo” kriterijus

- Paveiksle pavaizduota šaka negali vesti prie globalaus sprendinio. Kodėl?



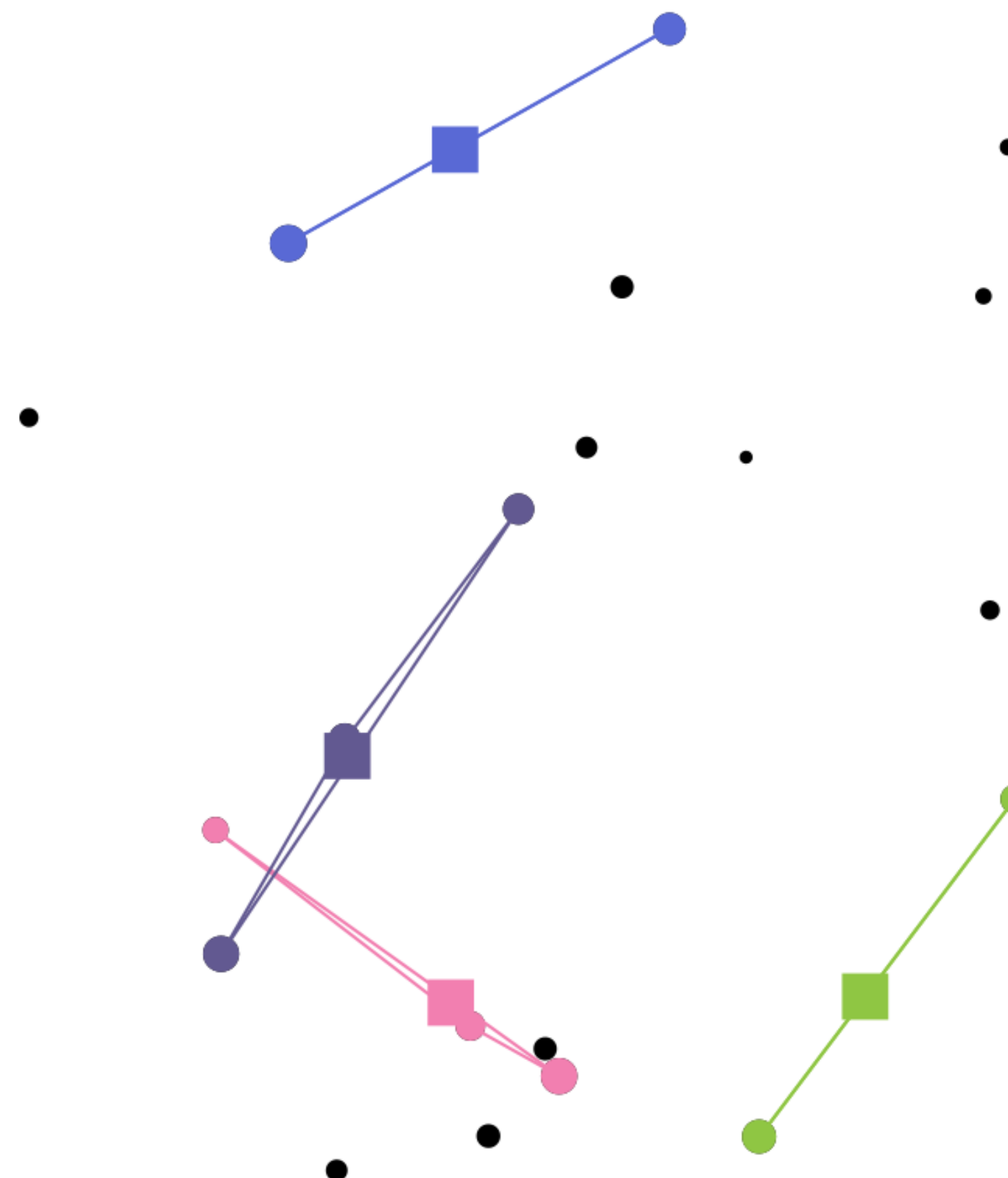
Perrankos algoritmas: šakų “pjovimo” kriterijus

- Paveiksle pavaizduota šaka negali vesti prie globalaus sprendinio. Kodėl?
 $\Rightarrow$ nes klasterių iškilieji apvalkalai privalo nekirsti vienas kito



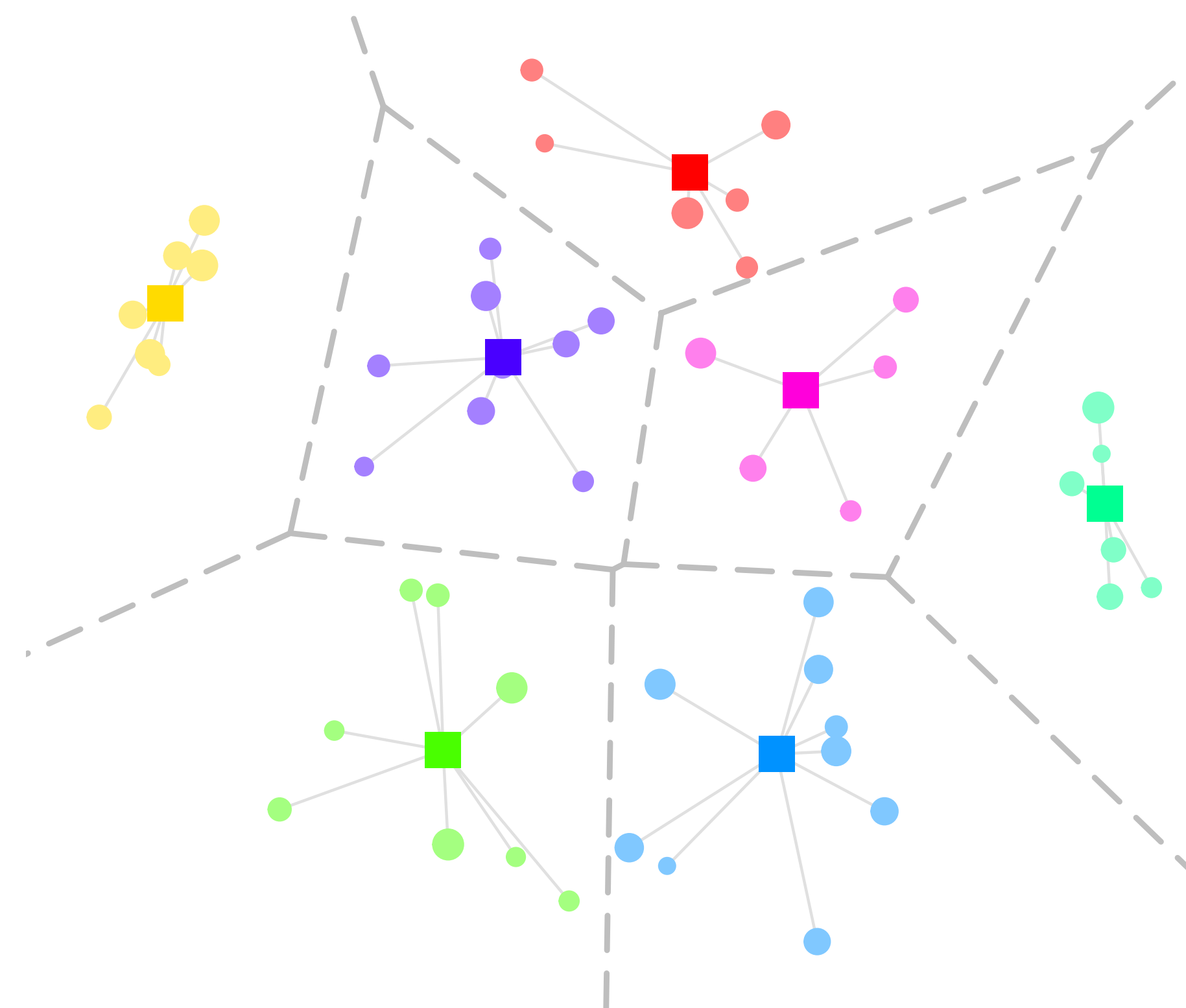
Perrankos algoritmas: šakų “pjovimo” kriterijus

- Paveiksle pavaizduota šaka negali vesti prie globalaus sprendinio. Kodėl?
 $\implies$ nes klasterių iškilieji apvalkalai privalo nekirsti vienas kito
- Įrodymas. Optimaliame grupavime, kiekvienas taškas priskiriamas artimiausiam centrui



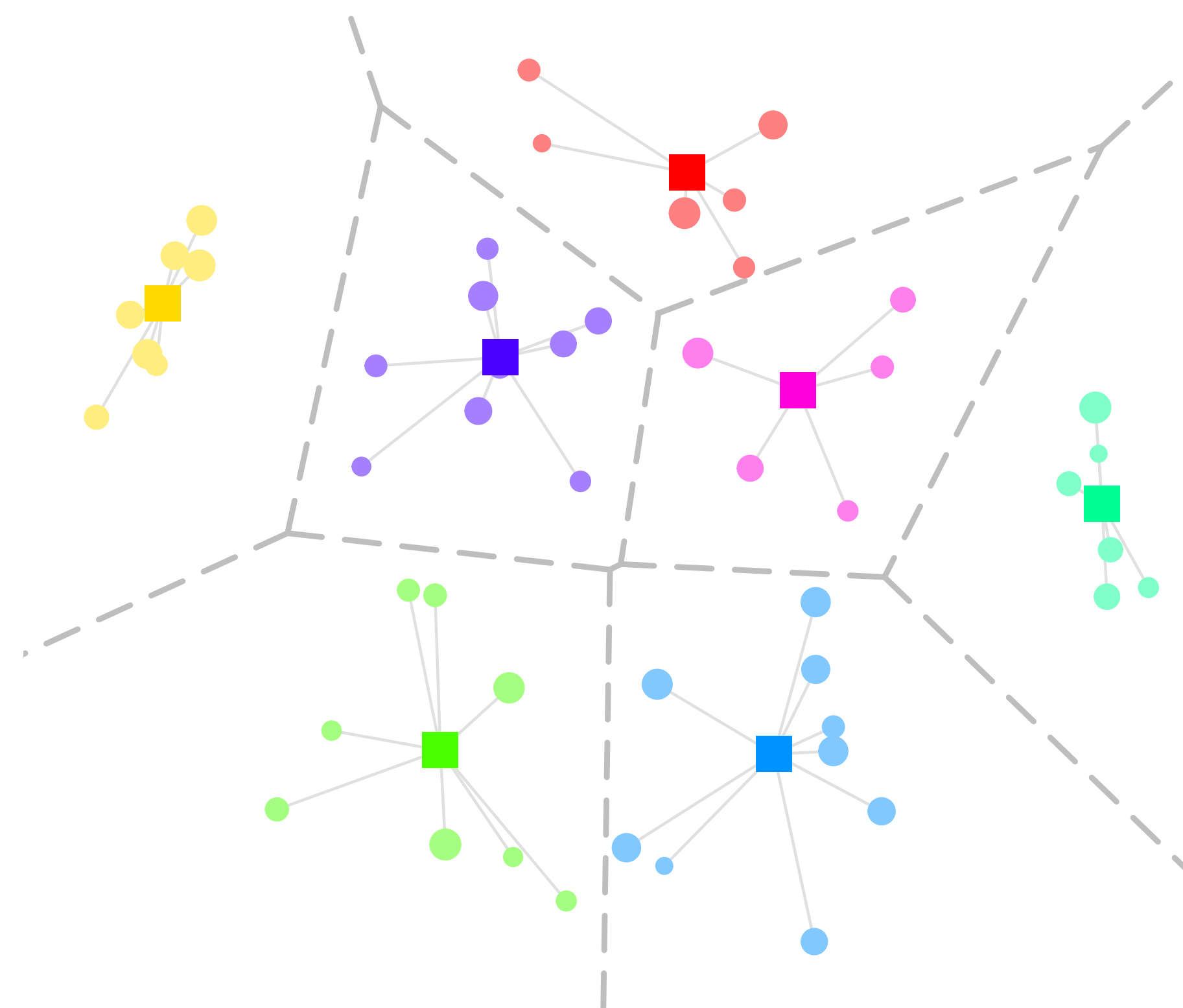
Perrankos algoritmas: šakų “pjovimo” kriterijus

- Paveiksle pavaizduota šaka negali vesti prie globalaus sprendinio. Kodėl?
 $\Rightarrow$ nes klasterių iškilieji apvalkalai privalo nekirsti vienas kito
- Įrodymas. Optimaliame grupavime, kiekvienas taškas priskiriamas artimiausiam centrui



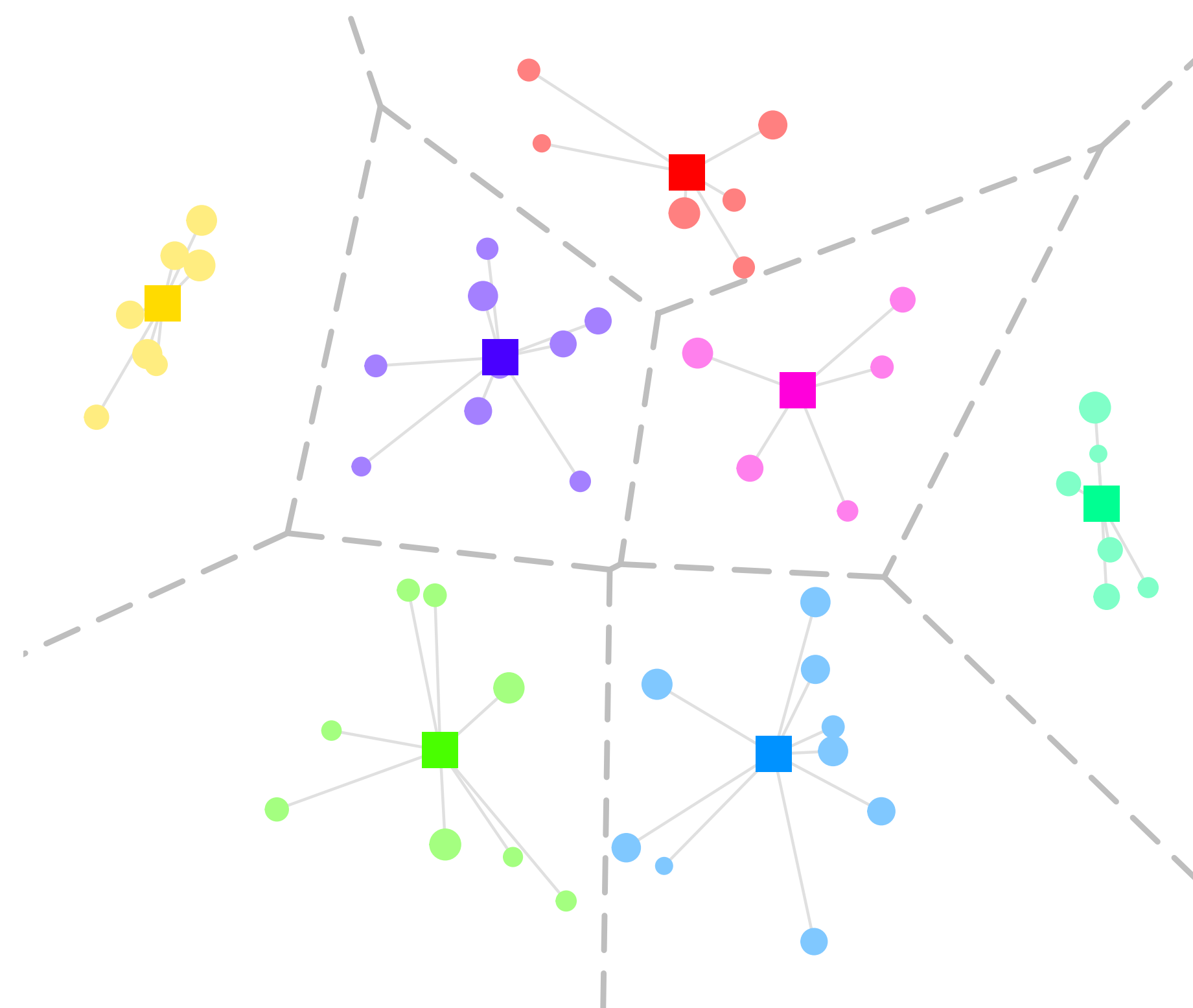
Perrankos algoritmas: šakų “pjovimo” kriterijus

- Paveiksle pavaizduota šaka negali vesti prie globalaus sprendinio. Kodėl?
 $\Rightarrow$ nes klasterių iškilieji apvalkalai privalo nekirsti vienas kito
- Įrodymas. Optimaliame grupavime, kiekvienas taškas priskiriamas artimiausiam centrui
 $\Rightarrow$ vadinasi, visi klasterio elementai pakliūna į šio centro Voronojaus ląstelę



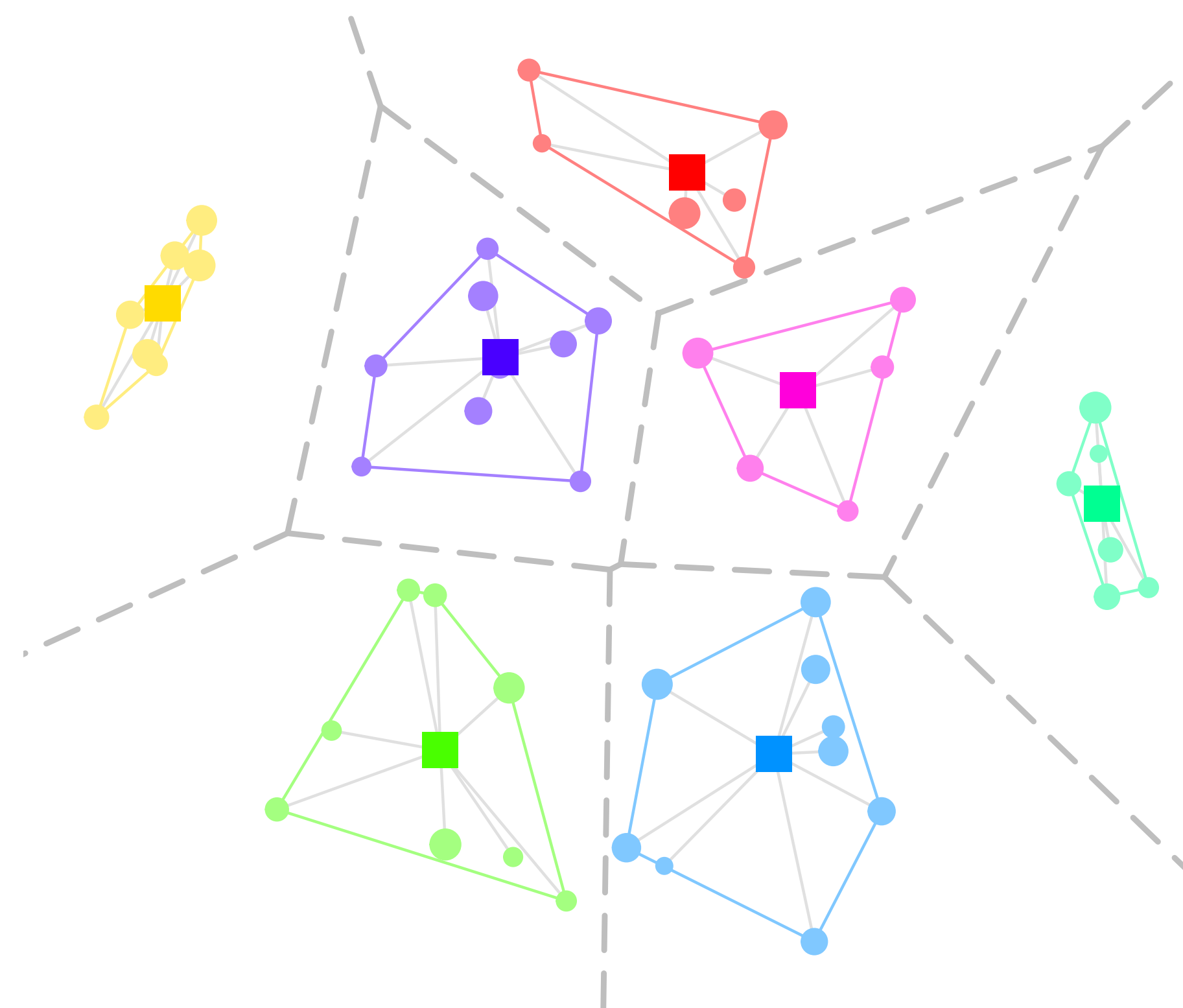
Perrankos algoritmas: šakų “pjovimo” kriterijus

- Paveiksle pavaizduota šaka negali vesti prie globalaus sprendinio. Kodėl?
 - ⇒ nes klasterių iškilieji apvalkalai privalo nekirsti vienas kito
- Įrodymas. Optimaliame grupavime, kiekvienas taškas priskiriamas artimiausiam centrui
 - ⇒ vadinasi, visi klasterio elementai pakliūna į šio centro Voronojaus ląstelę
 - ⇒ o taškų iškilasis apvalkalas yra šios Voronojaus ląstelės poaibis - taigi iškilieji apvalkalai optimaliame grupavime nesikerta



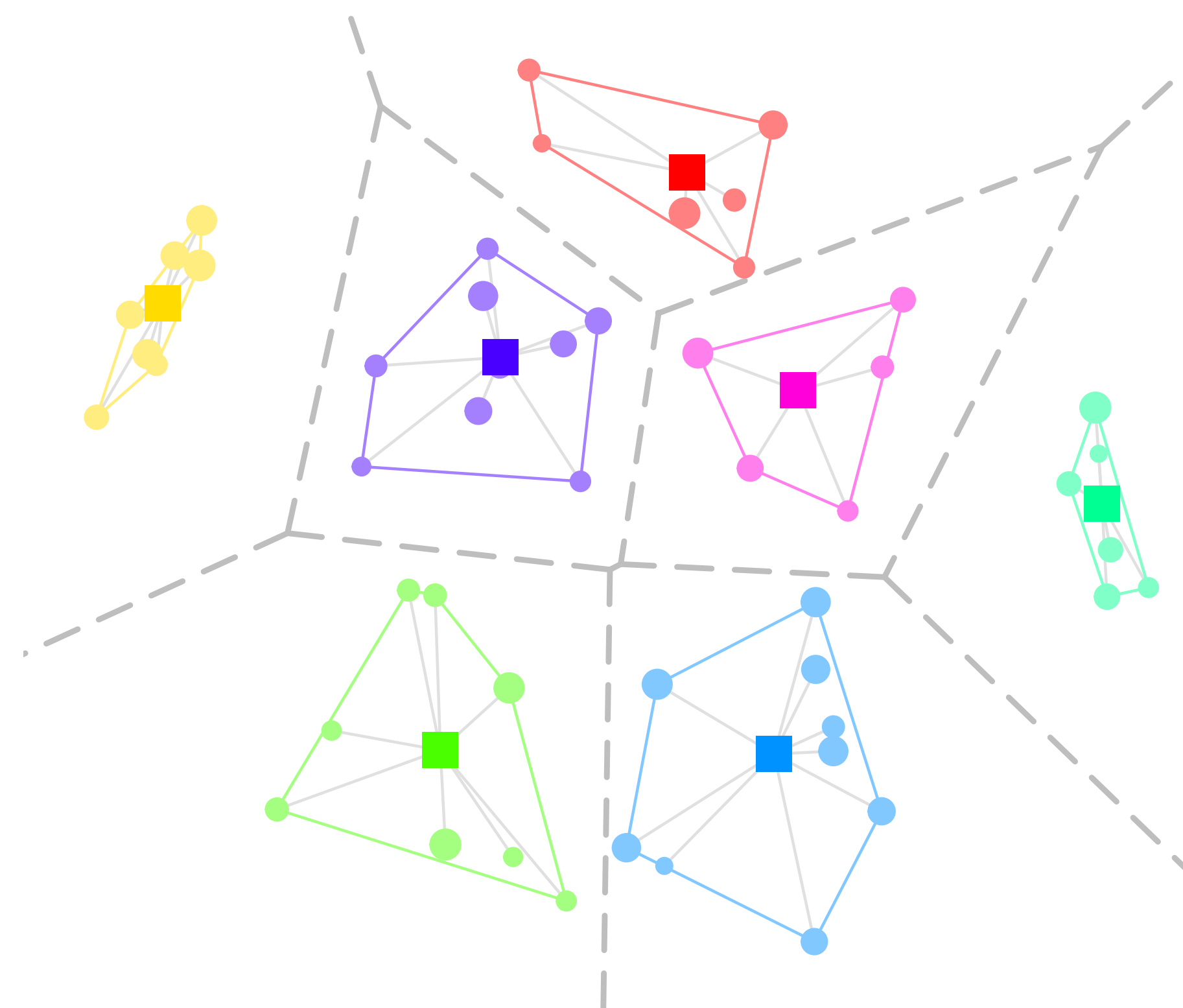
Perrankos algoritmas: šakų “pjovimo” kriterijus

- Paveiksle pavaizduota šaka negali vesti prie globalaus sprendinio. Kodėl?
 - ⇒ nes klasterių iškilieji apvalkalai privalo nekirsti vienas kito
- Įrodymas. Optimaliame grupavime, kiekvienas taškas priskiriamas artimiausiam centrui
 - ⇒ vadinasi, visi klasterio elementai pakliūna į šio centro Voronojaus ląstelę
 - ⇒ o taškų iškilasis apvalkalas yra šios Voronojaus ląstelės poaibis - taigi iškilieji apvalkalai optimaliame grupavime nesikerta



Perrankos algoritmas: šakų “pjovimo” kriterijus

- Paveiksle pavaizduota šaka negali vesti prie globalaus sprendinio. Kodėl?
 - ⇒ **nes klasterių iškilieji apvalkalai privalo nekirsti vienas kito**
- **Įrodymas.** Optimaliame grupavime, kiekvienas taškas priskiriamas artimiausiam centrui
 - ⇒ **vadinasi, visi klasterio elementai pakliūna į šio centro Voronojaus ląstelę**
 - ⇒ **o taškų iškilasis apvalkalas yra šios Voronojaus ląstelės poaibis - taigi iškilieji apvalkalai optimaliame grupavime nesikerta**
- Ši savybė galioja ir tuomet, jeigu centrų lokacijos yra apribotos.



Antras straipsnis: apžvalga

Antras straipsnis: apžvalga

- **Pirmo straipsnio uždavinys.** Rasti K klasterių $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_K$ su atitinkamais centrais $Q_1, Q_2, \dots, Q_K$, sprendžiančių uždavinį

$$\min_{Q_1, Q_2, \dots, Q_K} \sum_{k=1}^K \sum_{i \in \mathcal{C}_k} w_i \left\| P_i - Q_k \right\|_2^2 \quad \text{s.t.} \quad Q_k \in \mathcal{N}, k = 1, \dots, K$$

Antras straipsnis: apžvalga

- **Pirmo straipsnio uždavinys.** Rasti K klasterių $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_K$ su atitinkamais centrais $Q_1, Q_2, \dots, Q_K$, sprendžiančių uždavinį

$$\min_{Q_1, Q_2, \dots, Q_K} \sum_{k=1}^K \sum_{i \in \mathcal{C}_k} w_i \left\| P_i - Q_k \right\|_2^2 \quad \text{s.t.} \quad Q_k \in \mathcal{N}, k = 1, \dots, K$$

- **Antrame straipsnyje** spęsimė uždavinį

$$\min_{Q_1, Q_2, \dots, Q_K} \sum_{k=1}^K \sum_{i \in \mathcal{C}_k} w_i \left\| P_i - Q_k \right\|_2^1 \quad \text{s.t.} \quad Q_k \in \mathcal{N}, k = 1, \dots, K$$

Antras straipsnis: apžvalga

- **Pirmo straipsnio uždavinys.** Rasti K klasterių $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_K$ su atitinkamais centrais $Q_1, Q_2, \dots, Q_K$, sprendžiančių uždavinį

$$\min_{Q_1, Q_2, \dots, Q_K} \sum_{k=1}^K \sum_{i \in \mathcal{C}_k} w_i \left\| P_i - Q_k \right\|_2^2 \quad \text{s.t.} \quad Q_k \in \mathcal{N}, k = 1, \dots, K$$

- **Antrame straipsnyje** spręsimė uždavinį

$$\min_{Q_1, Q_2, \dots, Q_K} \sum_{k=1}^K \sum_{i \in \mathcal{C}_k} w_i \left\| P_i - Q_k \right\|_2^1 \quad \text{s.t.} \quad Q_k \in \mathcal{N}, k = 1, \dots, K$$

- Apibendrinsime perrankos algoritmą dimensijoms $d > 2$

Ačiū už dėmesį!

