

**Analiza comparativa între Algoritmi Genetici (GA) și Optimizarea Roiului de Particule
adaptata la cazul discret (DPSO) pentru Vertex Cover Problem (VCP)**

Gabriel Luican

Universitatea Transilvania Braşov, Facultatea de Matematica si Informatica

Automate, calculabilitate și complexitate

Lect. dr. Luciana Majercsik

Mai 2024

Autor: Gabriel Luican

Contact: gabriel.luican@gmail.com

Capitolul I - Problema Acoperirii Vârfurilor

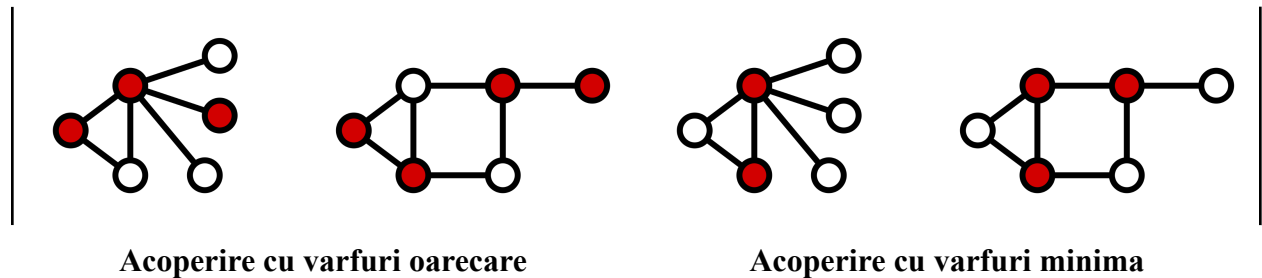
Descriere

În teoria grafurilor, o acoperire de vârf (uneori acoperire de noduri) a unui graf este un set de vârfuri care include cel puțin un punct final al fiecărei muchii a grafului.

În informatică, problema găsirii unei acoperiri minime de vârf este o problemă clasică de optimizare. Este NP-hard, deci nu poate fi rezolvat printr-un algoritm de timp polinomial dacă $P \neq NP$. În plus, este greu de aproximat – nu poate fi aproximat până la un factor mai mic de 2 dacă conjectura unică a jocurilor este adevărată. Pe de altă parte, are mai multe aproximări simple cu 2 factori. Este un exemplu tipic de problemă de optimizare NP hard care are un algoritm de aproximare. Versiunea sa de decizie, problema acoperirii vârfurilor, a fost una dintre cele 21 de probleme NP-complete ale lui Karp și, prin urmare, este o problemă clasică NP-completă în teoria complexității computaționale. În plus, problema acoperirii nodurilor este tratabilă cu parametri fix și o problemă centrală în teoria complexității parametrizate.

Aplicații

Optimizarea acoperirii nodurilor servește ca model pentru multe probleme din lumea reală și teoretice. De exemplu, o unitate comercială interesată să instaleze cât mai puține camere cu circuit închis posibil care să acopere toate holurile (marginile) care conectează toate camerele (nodurile) de pe un etaj ar putea modela obiectivul ca o problemă de minimizare a acoperirii vârfurilor. Problema a fost, de asemenea, folosită pentru a modela eliminarea secvențelor ADN repetitive pentru biologie sintetică și aplicații de inginerie metabolică.

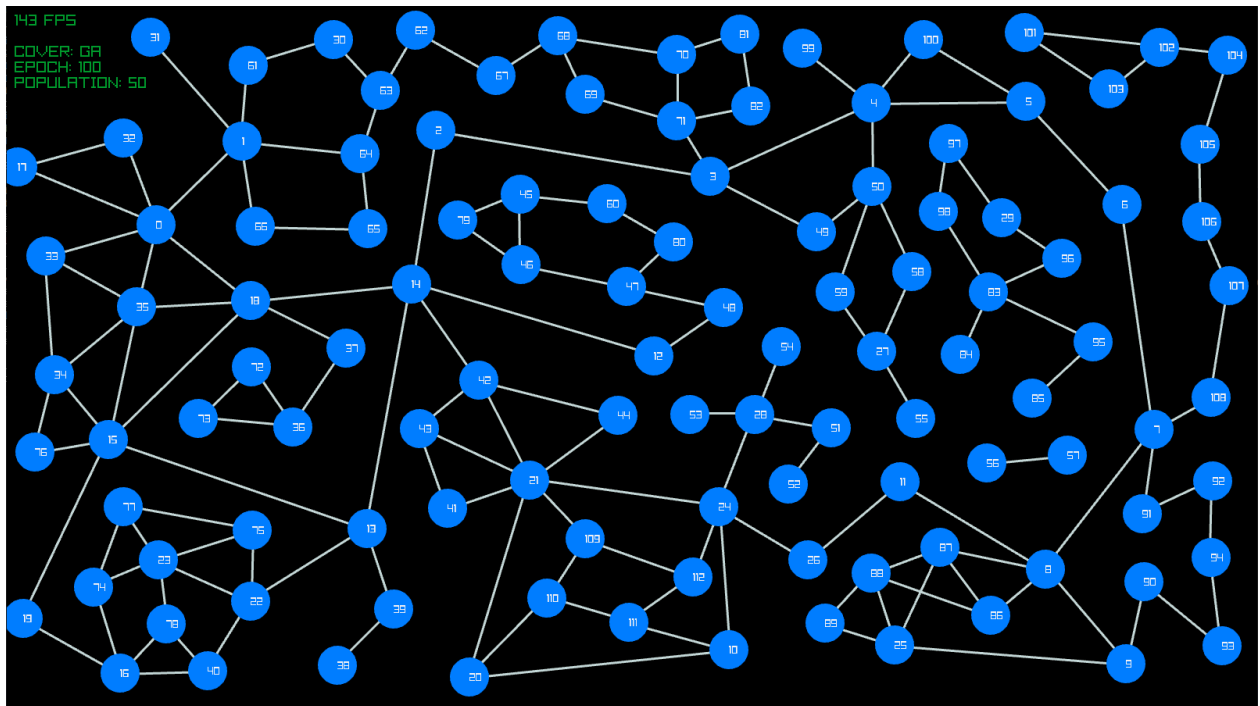


Relevanța

1. **Rețele de comunicații:** Problema acoperirii nodurilor poate fi utilizată pentru optimizarea plasării routerelor sau a punctelor de acces astfel încât fiecare conexiune sau segment de rețea să fie monitorizat sau controlat.
2. **Biologie computațională:** Acoperirea nodurilor poate ajuta la identificarea seturilor de proteine critice care interacționează cu alte proteine într-un mod esențial pentru funcționarea celulară.
3. **Inginerie software:** În testarea software-ului, problema acoperirii nodurilor poate fi folosită pentru a selecta un set minim de cazuri de testare care acoperă toate interacțiunile critice între componentele software-ului.
4. **Securitate:** Acoperirea nodurilor poate fi aplicată pentru a determina plasarea optimă a senzorilor sau a dispozitivelor de securitate pentru a monitoriza și proteja toate căile critice dintr-o infrastructură.
5. **Logistică și transport:** În problemele de logistică și transport, acoperirea nodurilor poate ajuta la optimizarea rutelor și plasarea centrelor de distribuție astfel încât toate locațiile importante să fie acoperite eficient.

Capitolul II - Implementare

Prezentare generală



VrtexCovrGL este un program scris în C++ care permite utilizatorilor să creeze și să vizualizeze grafuri, să aplice algoritmi genetici (GA) și optimizare prin roiuri de particule discrete (DPSO) pentru a rezolva problema acoperirii nodurilor (vertex cover). Programul dispune de o interfață grafică interactivă realizată cu ajutorul librăriei Raylib.

Caracteristici

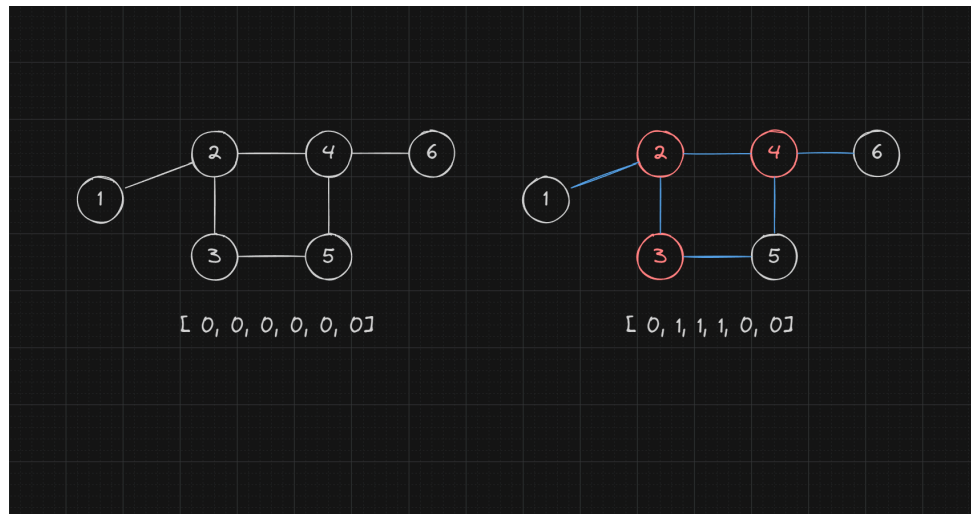
1. **Crearea nodurilor:** Click
2. **Formarea arcelor:** Click
3. **Conectare aleatorie a muchiilor:** Ctrl + A
4. **Inițializare algoritmi:** Ctrl + C
5. **Output:** Ctrl + D

- Programul generează un fișier cu timestamp care conține informații detaliate despre graf și cromozomi după rularea algoritmilor.
- Fișierul de output include timpii de execuție pentru fiecare algoritm și detalii despre starea finală a grafului și soluția obținută.

6. **Resetare graf:** Ctrl + F
7. **Initializare grafuri presetate:** Ctrl + 1, 2, 3, 4
8. **Comutare algoritmi:** Tab
9. **Calibrare epoci:** Ctrl + E + PLUS / MINUS
10. **Calibrare populație:** Ctrl + W + PLUS / MINUS

Metodologie GA

Când folosim algoritmul de evoluție genetică (GA), reprezentăm soluția problemei ca o secvență binară egală cu numărul de noduri. De exemplu, dacă avem un graf cu 6 noduri, o posibilă soluție pentru VCP ar putea arăta astfel:



Conform teoriei evoluției, dacă două organisme se împerechează, ele vor produce un descendent care va moșteni părți din ADN-ul lor și poate chiar va suferi mutații, dobândind "trăsături noi". Desigur, în viața reală, evoluția este mult mai complexă și are loc pe parcursul a miliarde de ani. Revenind la secvențele noastre binare, dacă le considerăm ADN, putem inițializa propriile "specii" și le putem reproduce. Creșterea lor va conduce la obținerea unei soluții pentru problema noastră.

În plus, vom adăuga un factor de "elitism", ceea ce înseamnă că vom păstra un număr predefinit de cei mai buni candidați pentru generația următoare, asigurând astfel că elitele sunt transmise mai departe.

Pași pentru GA

1. Inițializare a populației de cromozomi
2. Evaluare a fitness-ului
3. Selectie (Tournament selection)
4. Crossover (K-point)
5. Mutatie (Uniform)
6. Evaluare Minimului
7. Iterație a pașilor de selecție, crossover, mutatie, și evaluarea a minimului
8. Selectie finala

Metodologie DPSO

Discrete Particle Swarm Optimization (DPSO) este o extensie a binecunoscutului algoritm Particle Swarm Optimization (PSO) care abordează problemele de optimizare care implică variabile discrete.

Particle Swarm Optimization (PSO) este o tehnică de optimizare euristică inspirată de comportamentul social, în care o populație de particule (reprezentând soluții candidate) explorează în mod iterativ spațiul de căutare pentru a găsi soluții optime.

În PSO, fiecare particulă își ajustează poziția în funcție de propria experiență și de cea mai bună poziție găsită de întreg roiul.

Algoritmul original PSO presupune variabile continue, ceea ce limitează aplicabilitatea acestuia la problemele cu variabile discrete sau categorice.

Problemele de optimizare discretă implică variabile care pot lua numai valori specifice (de exemplu, numere întregi sau valori binare).

În loc să reprezinte soluțiile în mod direct, particulele reprezintă distribuții de probabilitate peste soluții posibile.

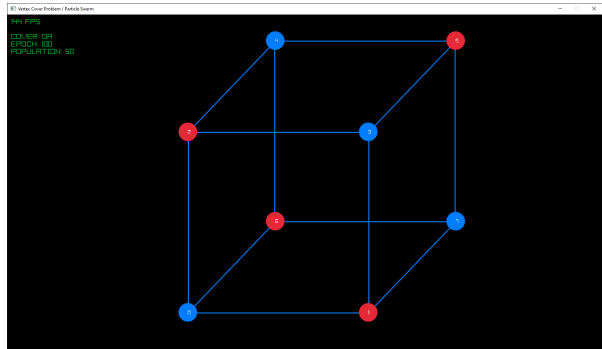
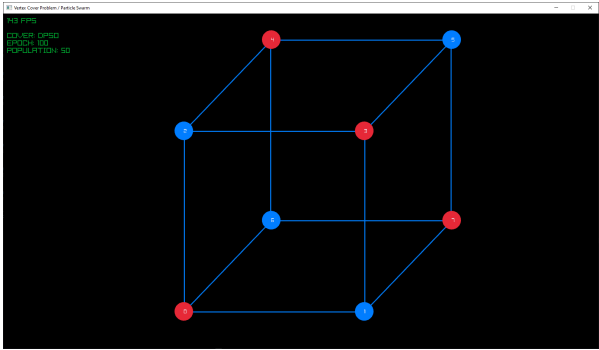
Actualizarea PSO modifică aceste distribuții de probabilitate, permițând particulelor să exploreze spații de soluții discrete.

Căutare în vecinătate: DPSO poate fi hibridizat cu metode de căutare locale pentru a îmbunătăți calitatea soluției.

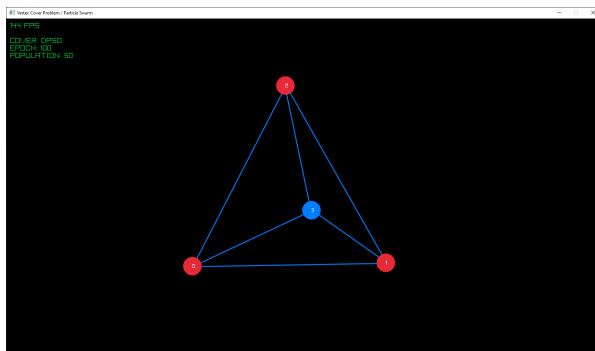
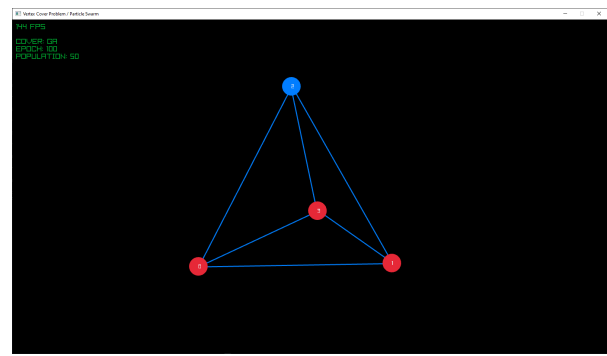
Pași pentru DPSO

1. Inițializare a populației de particule
2. Inițializare a celor mai bune poziții
3. Evaluare a fitness-ului
4. Actualizare viteze și poziții ale particulelor
5. Actualizare a celor mai bune poziții
6. Iterație a pașilor de evaluare a fitness-ului, actualizare a vitezelor și pozițiilor
7. Selecție finală

Prezentarea rezultatelor

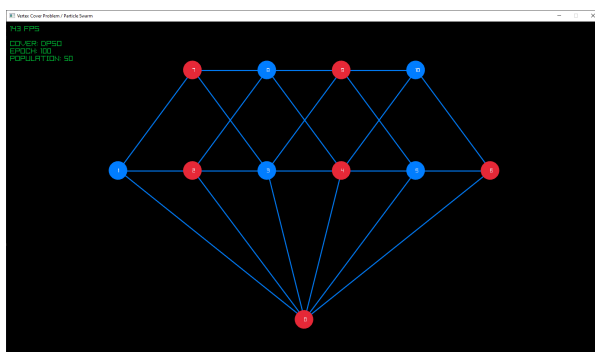
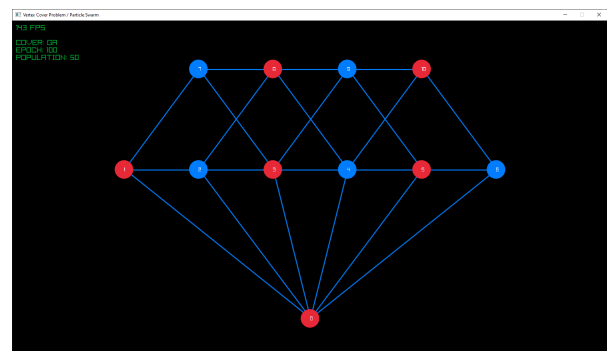
<pre>Nodes Count: 8 Edges Count: 12 Epochs: 100 Population: 50 Cover Type: GA Best Chromosome: 0 1 1 0 0 1 1 0 Best Solution: 4 Best Epoch: 3 Time Taken: 978ms</pre>	
	<pre>Nodes Count: 8 Edges Count: 12 Epochs: 100 Population: 50 Cover Type: DPSO Best Chromosome: 1 0 0 1 1 0 0 1 Best Solution: 4 Best Epoch: 1 Time Taken: 157ms</pre>

Nodes Count: 4
 Edges Count: 6
 Epochs: 100
 Population: 50
 Cover Type: GA
 Best Chromosome: 1 1 0 1
 Best Solution: 3
 Best Epoch: 0
 Time Taken: 684ms



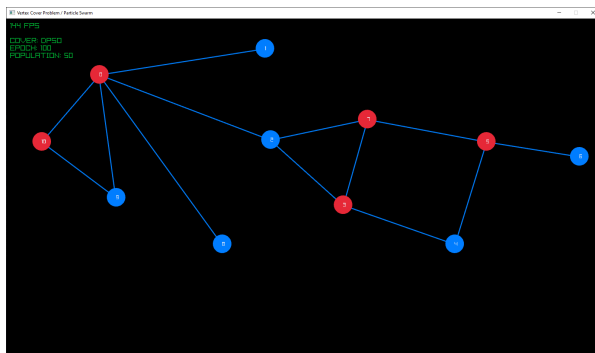
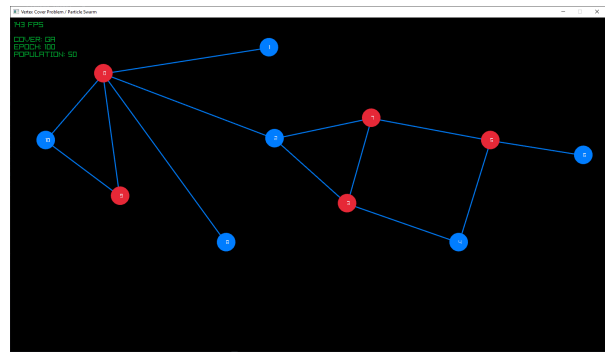
Nodes Count: 4
 Edges Count: 6
 Epochs: 100
 Population: 50
 Cover Type: DPSO
 Best Chromosome: 1 1 0 1
 Best Solution: 3
 Best Epoch: 0
 Time Taken: 102ms

Nodes Count: 11
 Edges Count: 22
 Epochs: 100
 Population: 50
 Cover Type: GA
 Best Chromosome: 1 1 0 1 0 1 0 0 1 0 1
 Best Solution: 6
 Best Epoch: 5
 Time Taken: 1261ms

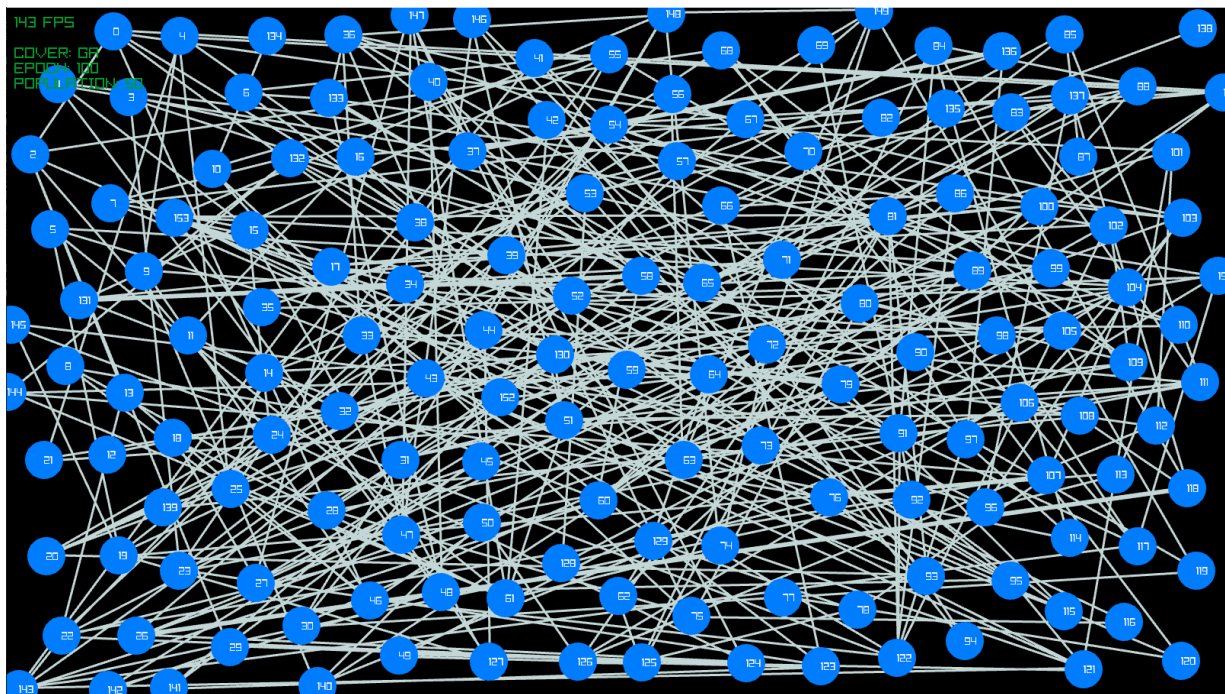


Nodes Count: 11
 Edges Count: 22
 Epochs: 100
 Population: 50
 Cover Type: DPSO
 Best Chromosome: 1 0 1 0 1 0 1 1 0 1 0
 Best Solution: 6
 Best Epoch: 8
 Time Taken: 207ms

Nodes Count: 11
Edges Count: 13
Epochs: 100
Population: 50
Cover Type: GA
Best Chromosome: 1 0 0 1 0 1 0 1 0 1 0
Best Solution: 5
Best Epoch: 5
Time Taken: 1055ms



Nodes Count: 11
Edges Count: 13
Epochs: 100
Population: 50
Cover Type: DPSO
Best Chromosome: 1 0 0 1 0 1 0 1 0 1 0
Best Solution: 5
Best Epoch: 2
Time Taken: 188ms



Nodes Count: 154
Edges Count: 528
Epochs: 500
Population: 100
Cover Type: GA
Best Chromosome: 0 1 1 0 0 1 1 0 1 1 0 0 0 1 1 1 0 1 1 1 0 0 1 0 1 0 1

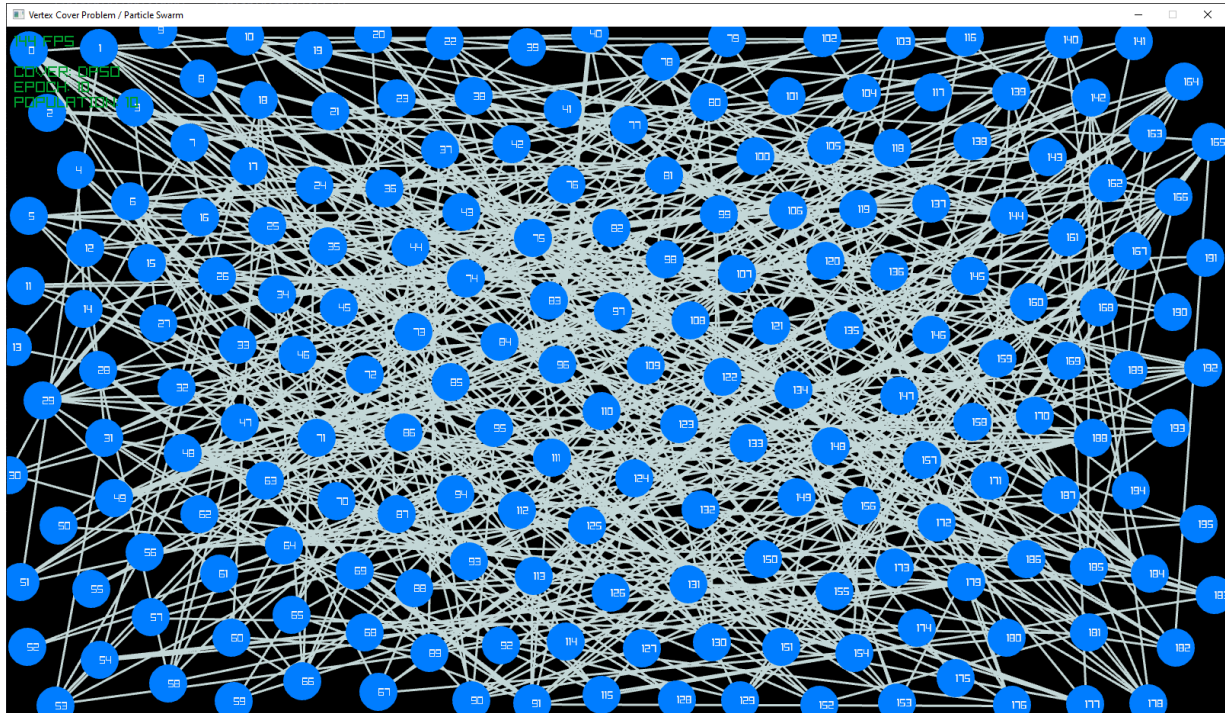
```
0 0 1 0 0 1 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 1 0 1 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 1 0
1 0 1 0 1 1 1 1 1 1 0 1 0 1 1 1 1 0 1 0 0 1 1 1 1 1 1 0 0 1 0 1 1 1 0 1 0 0 1 0 1 1
1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 0 0 0 0 0 1 1 1 1 0 1 1 1 1 1 0 1 1
```

Best Solution: 101
Best Epoch: 405
Time Taken: 1128126ms ~ 19 min

Nodes Count: 154
Edges Count: 528
Epochs: 500
Population: 100
Cover Type: DPSO

```
1 0 1 1 0 0 1 0 1 0 1 1 1 1 0 1 0 1 1 0 0 1 0 1 0 1 1 1 1 0 0 1 0 0 1 1 0 0 1 0 1 1
1 1 1 0 1 0 0 1 1 1 1 0 0 1 1 1 1 0 0 1 1 1 1 1 0 1 1 1 1 1 1 0 1 0 1 1 1 0 0 1 1 1
0 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1
0 0 0 0 1 1 1 1 1 0 0 1 0 1 1 1 1 0 1 0 1 1 0 1 1 0 1 1 0 1 0
```

Best fitness: 128420
Time Taken: N/A



Nodes Count: 196
Edges Count: 518
Epochs: 50
Population: 10
Cover Type: GA

Best Chromosome: 1 1 0 1 0 1 0 1 1 1 1 0 1 1 1 0 0 0 0 1 1 0 1 0 0 0 1 0 0 0 0 1 0 1
1 0 0 1 0 1 0 1 0 1 1 1 1 1 0 1 0 0 1 1 1 0 1 0 0 0 0 1 0 1 0 0 0 1 0 0 1 1 1 1 1 1
0 0 0 1 1 1 0 1 1 1 0 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 1 1 0 0 1 0 1 0 0 1 1 0
1 0 1 1 0 0 0 1 1 0 1 1 0 0 0 1 0 0 1 1 1 0 0 1 0 1 1 1 0 0 0 0 0 1 1 1 0 0 0 1 0 0
0 1 1 1 1 0 0 0 0 0 0 1 0 1 0 0 1 0 0 0 1 0 1 0 1 1 1 1 1 0 1 0 1 1 0

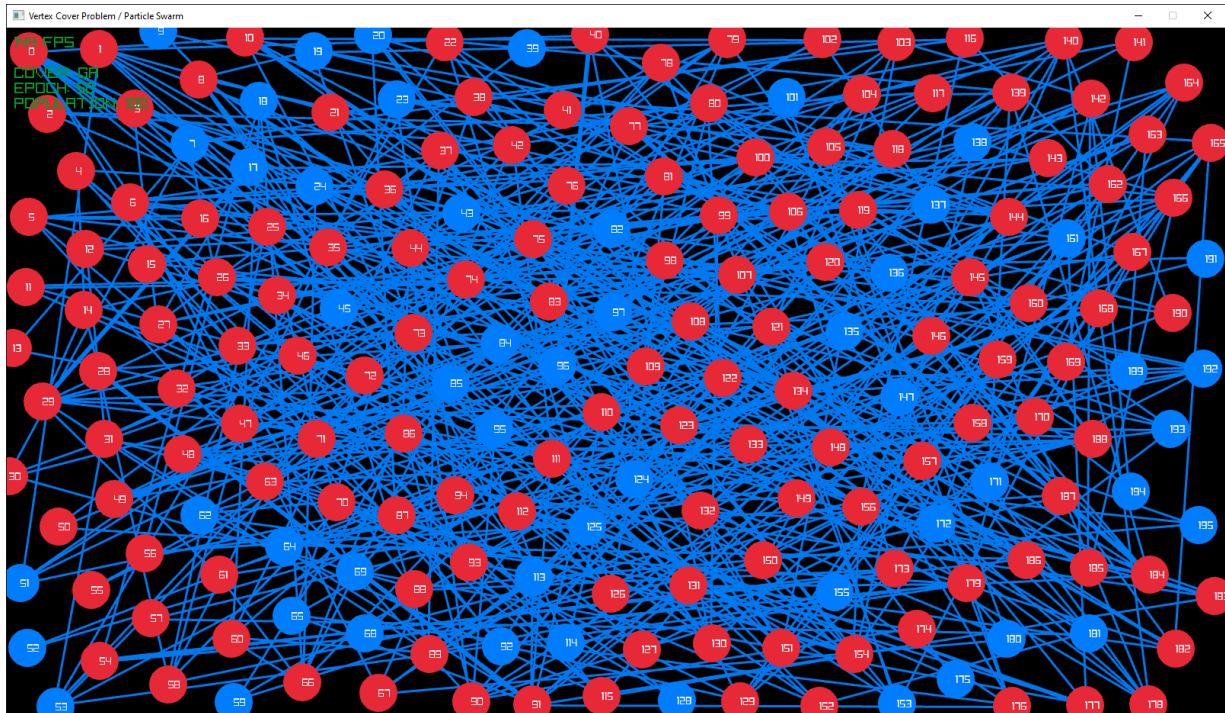
Best Solution: 400148

Best Epoch: 6
Time Taken: 878ms

Nodes Count: 196
Edges Count: 518
Epochs: 200
Population: 10
Cover Type: GA

Best Chromosome: 1 1 0 1 1 1 0 1 0 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 0 1 1 0 0 0 0 1 1
0 0 1 1 0 0 1 1 1 0 1 1 1 0 0 0 1 0 1 1 1 0 0 1 0 0 1 0 1 0 1 0 1 1 0 1 1 1 0 0 1 1
1 0 1 1 1 0 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 0 1 1 0 0 1 1 1 0 1 1 0 1 1 0 0 1 1 0 0
1 0 1 0 0 1 1 1 1 1 0 1 1 0 1 0 0 1 0 1 0 1 0 1 0 1 1 0 0 0 0 1 1 0 1 0 1 0 1 1 0 1
1 1 1 1 1 1 1 1 1 0 1 0 0 1 1 0 0 0 1 1 0 1 1 0 0 0 1 0 1 1 1 0 0 1 0 1 1

Best Solution: 234106
Best Epoch: 122
Time Taken: 3464ms



Nodes Count: 196
Edges Count: 518
Epochs: 50
Population: 100
Cover Type: GA

Best Chromosome: 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 0 0 0 0 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 0 1 1 1 0 1 0 1 1 1 1 1 0 0 0 1 1 1 1 1 0 1 1 0 1 0 0 1 1 0 0 1 1 1 1 1 1
1 1 1 1 1 1 1 0 1 0 0 1 1 1 1 1 1 0 1 1 0 0 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1
1 1 1 1 1 1 0 0 1 1 0 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 0 1 1 1 1
1 0 1 1 1 1 1 1 1 1 1 1 0 0 1 1 0 1 1 1 1 0 0 1 1 1 1 1 1 0 1 0 0 0 0 0 0

Best Solution: 144
Best Epoch: 47
Time Taken: 27987ms

