

Resolvendo o problema da Cobertura de Vértices utilizando o Simulated Annealing

Eduardo Miranda

3 de julho de 2018

Palavras-chave: Simulated Annealing, Vertex Cover Problem, Meta Heuristic

1 Introdução

O problema da cobertura de vértices consiste em, a partir de um grafo não direcionado determinar qual é a menor combinação de vértices selecionados que cubram todas as arestas do grafo. A resolução deste problema não é algo trivial, pois se faz necessário no pior dos casos testar todas as possibilidades possíveis de combinações de cobertura para encontrar-se a menor delas, não é surpresa o fato deste problema estar na classe dos NP-completos.

Uma aplicação seria achar os melhores lugares onde se colocar câmeras de vigilância em um prédio, ou até mesmo descobrir quais os melhores lugares para ser colocado antenas para que o sinal cubra toda uma cidade.

Tendo em mente que calcular uma solução por métodos exatos é na maioria das vezes inviável, opta-se por utilizar heurísticas, meta heurísticas ou métodos aproximados para sua resolução. Neste trabalho será utilizada a meta heurística do Simulated Annealing para a resolução deste problema em tempo hábil.

2 Metodologia

A meta heurística foi implementada na linguagem Python, utilizando a distribuição Linux Mint.

A representação da solução foi utilizada como um vetor permutações, que nada mais é do que um vetor de números inteiros que possui tamanho da quantidade de vértices do grafo passado. A partir desse vetor de permutações os vértices eram colocados na cobertura seguindo a ordem do vetor até que todas as arestas estivessem sido cobertas.

A meta heurística do Simulated Annealing tem algumas entradas essas seriam:

- Temperatura inicial
- Alpha

- Temperatura final
- Número de vizinhos gerados por iteração
- Número de reaquecimentos
- Solução inicial
- Método de gerar vizinhos

A temperatura inicial é utilizada para setar a temperatura atual toda vez que houver um reaquecimento.

O alpha é um multiplicador entre 1 e 0, que a cada iteração é multiplicado pela temperatura atual, fazendo com que a mesma diminua.

A temperatura final é o método de saída do simulated annealing, quando a temperatura atual é menor ou igual a temperatura final o laço é interrompido.

O número de vizinhos gerados é para cada iteração antes de se diminuir a temperatura quantos vizinhos devem ser gerados.

O número de reaquecimentos é o total de vezes em que a temperatura atual vai ser setada como temperatura inicial assim que atingir a temperatura final.

Como foi implementados dois métodos de ser gerada a solução inicial, sendo eles aleatoriamente e o guloso, este se torna um parâmetro para a resolução deste problema.

Para a geração de vizinhos também foi implementado dois métodos também, sendo eles o swap e a exclusão de vértices.

3 Resultados e Discussões

Os grafos utilizados para testes são provenientes do professor Diego que podem ser encontrado em [1].

Os resultados obtidos foram rodados com a seguinte configuração:

Temperatura inicial = 100;

Alpha = 0.9;

Temperatura final = 10;

Número de vizinhos gerados = 10;

Solução inicial = Aleatório ou Guloso;

Número de reaquecimentos = 7 ou 14;

Método de gerar vizinhos = Swap ou Exclusão de Vértice

Cada configuração foi rodada 100 vezes, variando o número de reaquecimentos, soluções iniciais e geração vizinhos, já as outras entradas eram fixas.

Melhor resultado do grafo	Melhor resultado obtido
53	56
50	52
55	57
54	58

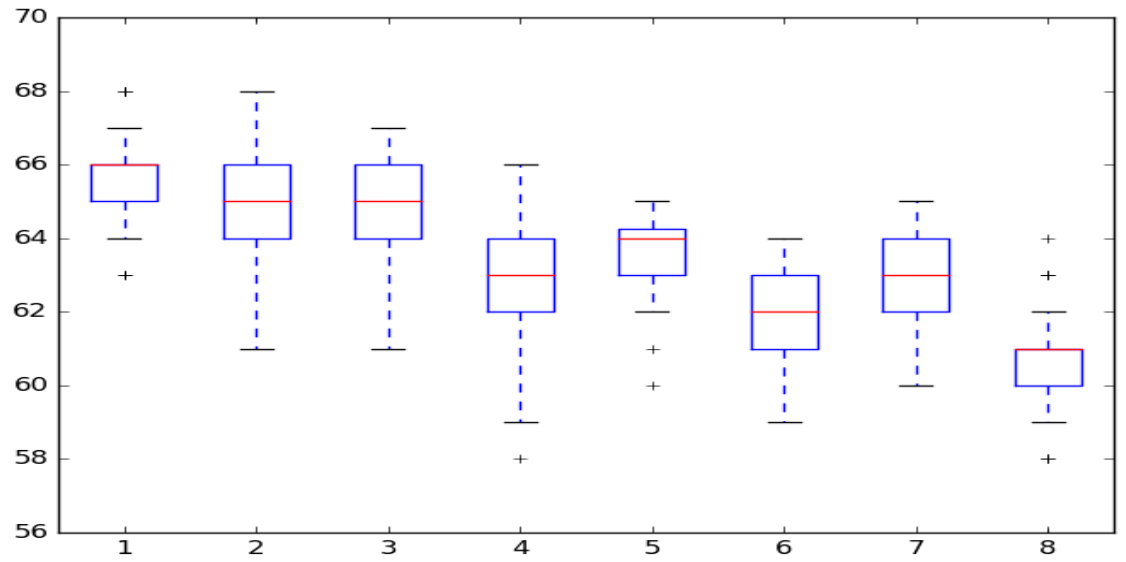


Figura 1: Resultado do grafo número 1

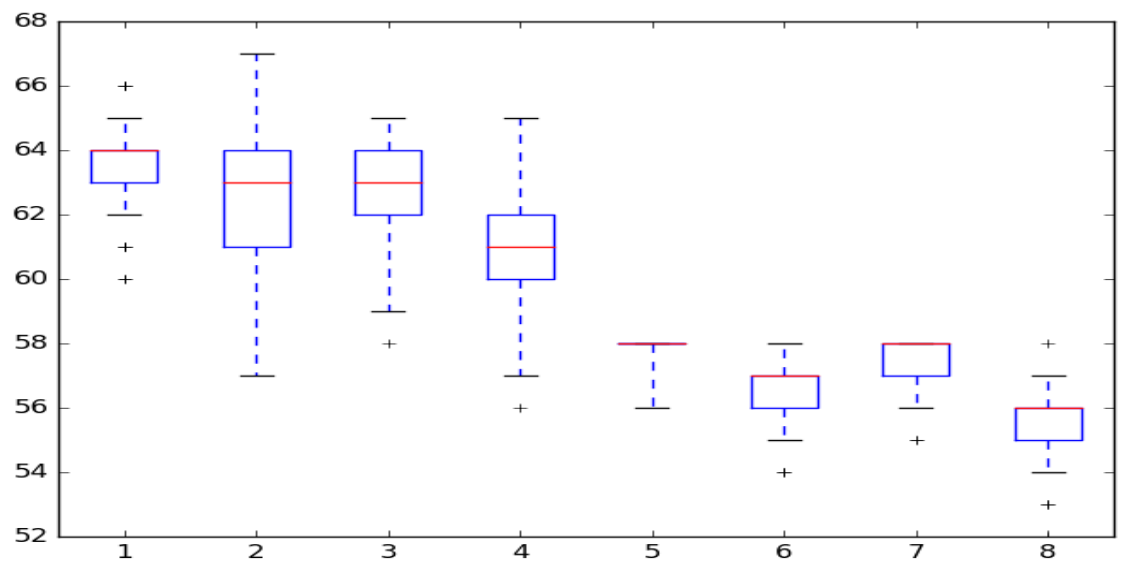


Figura 2: Resultado do grafo número 2

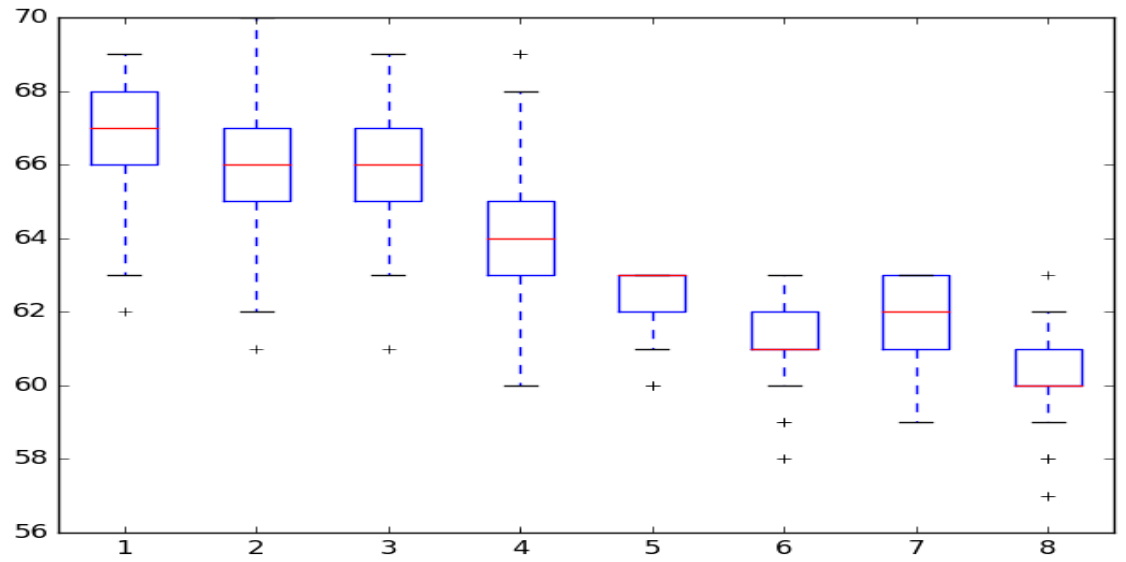


Figura 3: Resultado do grafo número 3

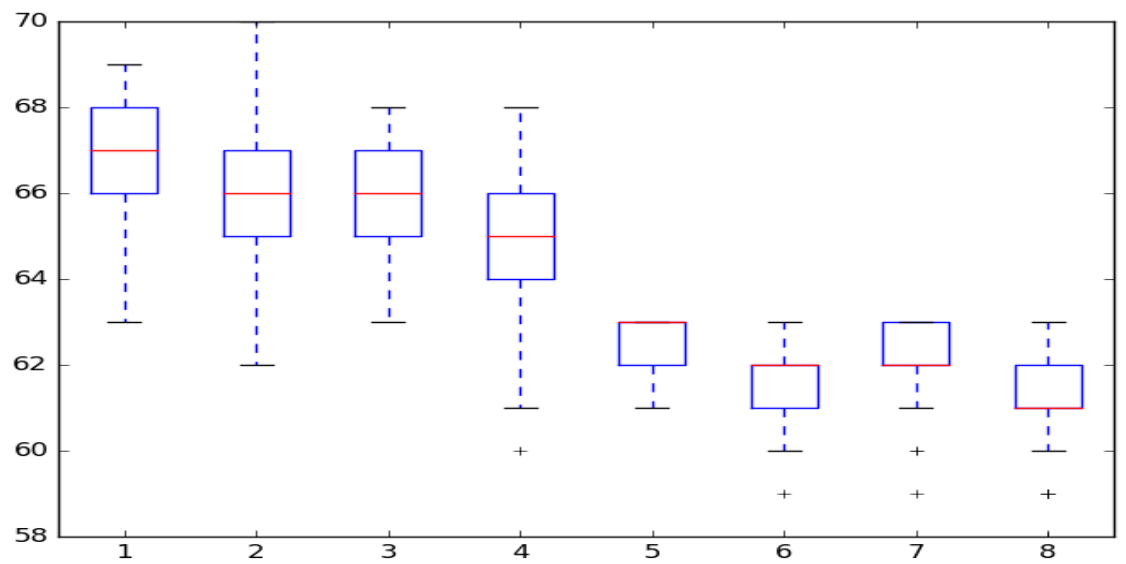


Figura 4: Resultado do grafo número 4

As combinações de 1 a 8 apresentadas nas imagens representam especificamente estas combinações:

1. Solução inicial aleatória + 7 reaquecimentos + gera vizinho por swap
2. Solução inicial aleatória + 7 reaquecimentos + gera vizinho por exclusão
3. Solução inicial aleatória + 14 reaquecimentos + gera vizinho por swap
4. Solução inicial aleatória + 14 reaquecimentos + gera vizinho por exclusão
5. Solução inicial gulosa + 7 reaquecimentos + gera vizinho por swap
6. Solução inicial gulosa + 7 reaquecimentos + gera vizinho por exclusão
7. Solução inicial gulosa + 14 reaquecimentos + gera vizinho por swap
8. Solução inicial gulosa + 14 reaquecimentos + gera vizinho por exclusão

Abaixo estão algumas estatísticas mais detalhadas sobre os resultados dos grafos 1 a 4:

Tabela 1: Estatísticas do resultado do grafo número 1

Sol inicial	Qtd. reaquecimentos	Geração de vizinhos	Média	Melhor encontrado	Coef. de variação
Aleatória	7	Swap	65.76	63	1.1886
Aleatória	7	Exclusão de vértice	65.19	61	1.1509
Aleatória	14	Swap	64.75	61	1.1509
Aleatória	14	Exclusão	62.83	58	1.0943
Gulosa	7	Swap	63.85	60	1.1320
Gulosa	7	Exclusão	62.06	59	1.1132
Gulosa	14	Swap	63.09	60	1.1320
Gulosa	14	Exclusão	60.49	56	1.0566

Tabela 2: Estatísticas do resultado do grafo número 2

Sol inicial	Qtd. reaquecimentos	Geração de vizinhos	Média	Melhor encontrado	Coef. de variação
Aleatória	7	Swap	63.46	60	1.2
Aleatória	7	Exclusão de vértice	62.46	57	1.14
Aleatória	14	Swap	62.6	58	1.16
Aleatória	14	Exclusão	60.67	56	1.12
Gulosa	7	Swap	57.77	56	1.12
Gulosa	7	Exclusão	56.55	54	1.08
Gulosa	14	Swap	57.43	55	1.1
Gulosa	14	Exclusão	55.62	52	1.04

Tabela 3: Estatísticas do resultado do grafo número 3

Sol inicial	Qtd. reaquescimentos	Geração de vizinhos	Média	Melhor encontrado	Coef. de variação
Aleatória	7	Swap	66.68	62	1.1272
Aleatória	7	Exclusão de vértice	66.04	61	1.1090
Aleatória	14	Swap	65.8	61	1.1090
Aleatória	14	Exclusão	64.23	56	1.0181
Gulosa	7	Swap	62.38	60	1.0909
Gulosa	7	Exclusão	61.23	58	1.0545
Gulosa	14	Swap	61.69	59	1.0727
Gulosa	14	Exclusão	60.3	57	1.0363

Tabela 4: Estatísticas do resultado do grafo número 4

Sol inicial	Qtd. reaquescimentos	Geração de vizinhos	Média	Melhor encontrado	Coef. de variação
Aleatória	7	Swap	66.99	63	1.1666
Aleatória	7	Exclusão de vértice	66.29	62	1.1481
Aleatória	14	Swap	65.93	63	1.1666
Aleatória	14	Exclusão	64.51	60	1.1111
Gulosa	7	Swap	62.63	61	1.1296
Gulosa	7	Exclusão	61.71	59	1.0925
Gulosa	14	Swap	62.26	59	1.0925
Gulosa	14	Exclusão	61.15	58	1.0740

4 Conclusão

A configuração que obteve as melhores soluções visivelmente mostrado nos gráficos e tabelas foi a número 8. Isto ocorreu pelo fato dela começar com uma solução inicial melhor e utilizar um método de vizinhança mais inteligente, que é tirar um vértice que já está na cobertura.

Pode concluir-se que o Simulated Annealing é método robusto para obtenção de resultados. Mesmo quando não é colocado alguma inteligência na hora de fazer a codificação, ele gera resultados que são aproximadamente 12% piores que o melhor conhecido, porém quando alguma informação sobre o problema é considerada na codificação os resultados ficam em torno de 7%.

Referências

- [1] “Especificação 2016-2,” <https://docs.google.com/viewer?a=v&pid=sites&srcid=aWZtZy5lZHUuYnJ8ZGllZ29zaWx2YXxneDo0MjdlODkwNDIxNTcwNDYw>, acessado em 03/07/2018.