Trabalho de Simulação AD 2021/1
Simulando Redes Robustas e Eficientes

15 de outubro de 2021

Resumo

O propósito deste trabalho é experimentar com MCMC e simulação de eventos discretos, aproveitando para recordar algoritmos em grafos.

1 Data e formato de entrega

Os trabalhos podem ser em grupos de até no máximo 4 alunos.

Entregar relatório em PDF e código em github, dia 22 de outubro.

Fazer apresentação de 15-20 minutos, nas aulas dos dias 18 e 20 de outubro, com alguns poucos slides. O propósito das apresentações é indicar desafios e ter tempo de possivelmente fazer ajustes finais até entrega.

2 Introdução e Objetivos

Robustez e eficiência são duas das características desejáveis em sistemas computacionais. Em redes de computadores, por exemplo, robustez refere-se ao fato de as redes serem tolerantes a falhas, e eficiência refere-se ao fato de os caminhos entre os nós serem curtos. A busca por topologias de redes robustas e eficientes motivou inúmeros trabalhos na interseção entre redes de computadores, teoria dos grafos e inteligência artificial.

Embora robustez e eficiência sejam ambas desejáveis, pode haver um compromisso entre os dois. Considere, por exemplo, que exista um custo para adicionar uma conexão entre dois nós dentro de uma rede. Assuma que o orçamento para uma rede com n vértices só comporte $m = n + 1$ conexões diretas (arestas). Nesse caso, é possível alcançar eficiência máxima, por exemplo, conectando todos os demais vértices por meio de um nó central (semelhante a uma forma de estrela), envolvendo $n - 1$ arestas, e as duas arestas adicionais sendo conectadas arbitrariamente, conferindo uma distância entre nós sempre menor ou igual a 2. Em outras palavras, a distância entre nós é $O(1)$. Entretanto, esta topologia derivada da estrela é muito frágil, tendo em vista que a remoção do nó central desconecta os nós restantes. Alternativamente,

pode-se ter uma solução muito robusta, contendo um ciclo envolvendo todos os nós, na qual é impossível desconectar quaisquer dois nós com a remoção de um terceiro. Em contrapartida, nesse caso a distância entre os nós será da ordem de $O(n)$. A topologia derivada a partir do ciclo é muito mais robusta, mas ao mesmo tempo muito menos eficiente em comparação com a topologia derivada a partir da estrela.

3 Objetivos

Este trabalho tem dois objetivos: 1) encontrar redes robustas e eficientes, usando MCMC e 2) entender como que essas redes se comportam frente a falhas e reparos.

- **MCMC:** visamos responder a seguinte pergunta: restringindo-se a redes robustas, ou seja, biconexas, quais as topologias mais eficientes? Surpreendentemente, embora esta pergunta seja muito simples, sua resposta ainda não é conhecida.
- **falhas e reparos:** assumindo que falhas ocorrem segundo um fluxo Poisson com taxa λ e reparos ocorrem segundo um fluxo Poisson com taxa μ, qual a fração de tempo em que a rede está desconexa?

4 MCMC (Cadeia de Markov Tempo Discreto)

Nesta parte do trabalho você tem flexibilidade para desenvolver o algoritmo MCMC como julgar mais adequado.

As entradas do programa são:

1. o número de vértices, n
2. o número de arestas, m, que deve ser necessariamente $n + 1 \leq m \leq 2n - 6$

A saída do programa deve ser um ou mais grafos, biconexos, tais que a soma das distâncias entre os vértices seja a menor possível.

Sinta-se a vontade para experimentar com diferentes valores de m e n, mas o seu trabalho deve contemplar, ao menos, os seguintes casos

1. $n = 5, m = 6$ (nesse caso, excepcionalmente, estamos permitindo $m > 2n - 6$)
2. $n = 5, m = 7$ (nesse caso, excepcionalmente, estamos permitindo $m > 2n - 6$)
3. $n = 16, m = 20$
4. $n = 20, m = 25$
5. $n = 28, m = 31$
6. $n = 29, m = 30$

7. $n = 32, m = 50$

8. $n = 32, m = 60$ (bonus, excepcionalmente permitindo $m > 2n - 6$)

5 Falhas e Reparos (Cadeia de Markov Tempo Contínuo)

Dada uma rede com n vértices e m arestas, considere agora os seguintes cenários

1. taxa de falha por link igual a λ, taxa de reparo por link igual a μ

2. taxa de falha por link igual a λ, taxa de reparo global, para todos os links, igual a μ (ou seja, na medida em que os links falham, eles entram em uma fila, que é servida de forma FCFS, e na qual o servidor tem capacidade de serviço μ)

No cenário 1, cada link é reparado, na média, a cada $1/\mu$ unidades de tempo. No cenário 2, a cada $1/\mu$ unidades de tempo inicia-se um reparo de um novo link, sendo que o link a ser reparado é escolhido de forma FCFS dentre os links que estão na fila aguardando para serem reparados.

Resolva os seguintes casos

1. escolha 5 dos 7 cenários acima, pegue a rede mais eficiente que você encontrou em cada um deles, e rode uma simulação para cada um dos cenários acima, considerando os seguintes dois casos: tempo de reparo exponencialmente distribuído e tempo de reparo distribuído de forma determinística

Em cada caso, reporte

1. a fração de tempo em que o sistema fica desconectado

2. a fração de tempo em que pelo menos um link está falho

3. a fração de tempo em que o sistema está conectado mas, ainda assim, tem ao menos um link falho

Assuma que, enquanto o link está falho ou sob reparo, ele fica indisponível. A rede está desconectada se ao menos um par de nós não consegue se comunicar um com o outro (ou seja, se não existe caminho entre eles).

Assuma $\lambda = 1$ e $\mu = 1$.

6 Simulação para Verificar Resultados Analíticos

Considere uma rede bem simples, por exemplo, com 5 vértices, e 6 arestas. Resolva o problema do item anterior de duas formas distintas: 1) construindo a cadeia de Markov, e resolvendo-a e 2) via simulação por eventos discretos. Mostre que a solução obtida via cadeia de Markov está dentro do intervalo de confiança da simulação.

7 Criatividade

Use sua criatividade para resolver variantes do problema proposto, explorar outros parâmetros, indicar extensões de trabalhos futuros etc.

8 Como Resolver uma CMTC no Matlab

A seguir, ilustramos como resolver a cadeia com a seguinte matriz de taxas no Matlab.

$$Q = \begin{pmatrix} -1 & +1 \\ +2 & -2 \end{pmatrix} \tag{1}$$

```
>> Q=[-1,1;2,-2]

Q =

    -1     1
     2    -2

>> Qt=Q'

Qt =

    -1     2
     1    -2

>> s=size(Qt,1)

s =

     2

>> Qt(2,:)=1

Qt =

    -1     2
     1     1


>> e=zeros(s,1)

e =
```

```
     0
     0

>> e(s)=1

e =

     0
     1



>> Qt\e

ans =

    0.6667
    0.3333
```

9 Quesitos Que Devem Constar no Relatório

1. Desafios encontrados e soluções
2. Todas as soluções obtidas via simulação devem vir acompanhadas de seus intervalos de confiança
3. Decisões de projeto: breve descrição do simulador, e de como foi implementado
4. Depuração (como o simulador foi depurado). Em particular
   (a) quando possível, comparar analítico com simulação e verificar que analítico está no intervalo de confiança da simulação
   (b) simplificar o problema, e estudar casos básicos
   (c) colocar *printf* no código para rastrear o que ocorre ao longo do processo
5. Resultados
6. Discussão dos resultados

10 Como trabalho será avaliado

O trabalho será avaliado com relação a corretude, completude e criatividade.

1. MCMC: 40 pontos
2. CMTC: 40 pontos (10 pontos analítico, 20 pontos simulação, 10 pontos mostrando que analítico corresponde à simulação)
3. cenários adicionais/criativos: 20 pontos

Em todos os passos, mostre que os seus resultados estão corretos (usando depuração adequada e indicando as estratégias usadas para depurar).

A pontuação leverá em conta a apresentação e o relatório final.

11 O que entregar?

1. relatório em PDF
2. código fonte no github, colab, ou similar.
3. slides, a serem apresentados em aula (15-20 mins de apresentação)

12 Comentários adicionais

Para calcular a soma das distâncias entre os vértices você pode usar o algoritmo de all-pairs-shortest-path `https://en.wikipedia.org/wiki/Floyd%E2%80%93Warshall_algorithm`

Para determinar se um grafo é conexo ou não, você pode usar algoritmos clássicos de grafos. Idem, para dterminar se um grafo é biconexo.

Você não precisa implementar, do zero, nenhum dos algoritmos acima. Você pode pegar implementações prontas, mas precisa documentar todas as fontes que usar.

13 Importante! Assista esses vídeos antes de começar o trabalho

É fundamental que você assista esta playlist:

`https://www.youtube.com/playlist?list=PLVRCSkl8sA_mmx1J50BCLUkhU46HPgJa2`