

Trabalho Prático 2 - Problema de Alocação de Canais em redes de Telefonia Celular

Esse trabalho prático tem como objetivo a implementação de uma solução ótima e uma aproximada para o problema de alocação de canais em redes de telefonia celular, cujo objetivo é minimizar o número de canais necessários. Será valorizado com ponto(s) extra(s) para quem apresentar uma análise teórica do fator de aproximação da heurística implementada.

Problema

Você foi contratado(a) pela *previousTel*, uma nova empresa de telefonia móvel em Belo Horizonte, cujo objetivo é prover serviços de baixo custo para a região metropolitana. O serviço de telefonia móvel funciona da seguinte maneira: existe um espectro de frequências reservado para o uso em telefonia móvel que por sua vez é dividido em canais. Estes canais, no Brasil, são regulados pela *Anatel* que é responsável pela venda/concessão de suas licenças de uso. Como dito anteriormente, a *previousTel* é uma empresa nova que possui uma grande limitação orçamentária, portanto pretende-se comprar o menor número possível de licenças para seu pleno funcionamento. Na prática, a empresa, pretende alocar apenas um canal para cada uma de suas Estações Rádio Base (*ERBs*). Como também, não há verba para a criação de suas próprias *ERBs*, um pedido foi encaminhado ao Estado, requisitando uma licença temporária de uso das *ERBs* construídas por uma antiga empresa estatal de telefonia que também atuava em Belo Horizonte. Felizmente, o estado, entendendo que seria economicamente interessante o surgimento de uma nova empresa para competir com as grandes telefonias já consolidadas, concedeu o pedido.

Seu problema agora consiste-se em, dado um conjunto de *ERBs* com um determinado raio de atuação, determinar o menor número de licenças de canais que devem ser comprados, de modo que *ERBs* que possuem áreas de cobertura comum (possibilidade de interferência) não sejam alocadas para funcionarem no mesmo canal.

Pede-se que este problema seja modelado usando grafos, onde cada *ERB* seja representada por uma vértice e as arestas indiquem que duas antenas possuem áreas de atuação em comum, possibilitando interferência. Lembre-se que duas soluções devem ser apresentadas, a ótima e a aproximada (gerada por qualquer heurística).

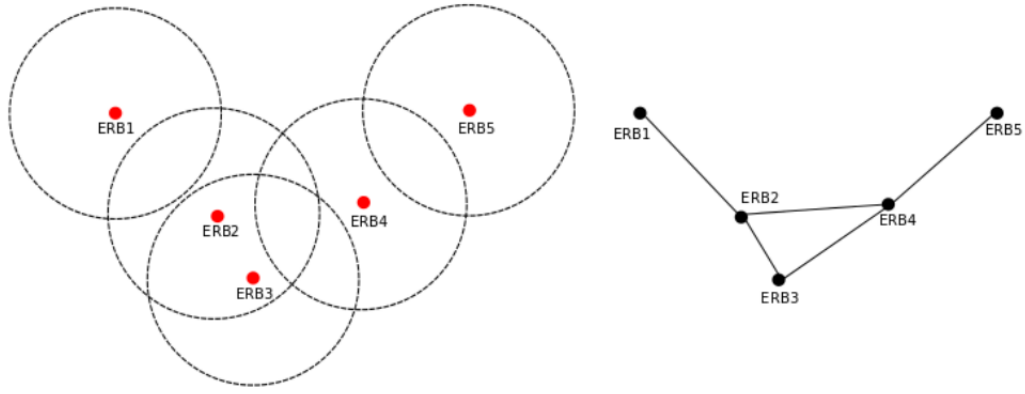


Figura 1: Exemplo de grafo representativo da interferência entre ERBs

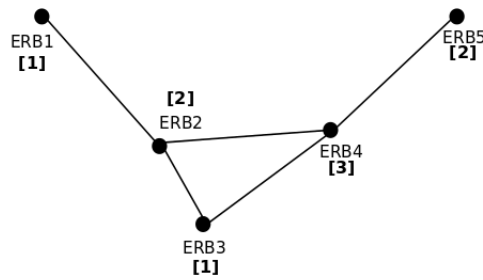


Figura 2: Exemplo de alocação de canais[1,2,3] de rede de Telefonia Celular

Entrada e Saída

Seu *Makefile* deverá gerar dois executáveis: um para a solução aproximada (`./tp2h`) e outro para a ótima (`./tp2o`). Cada um destes programas, deverá solucionar múltiplas instâncias do problema em uma única execução. Serão passados na entrada de dados informações para a construção do grafo que representará a rede de telefonia celular da cidade de Belo Horizonte e Região Metropolitana. A saída será dada pelo número mínimo de canais utilizados para cobrir a alocação de todas as antenas. A entrada será lida de um arquivo e o resultado do programa deve ser impresso em outro arquivo de saída. Ambos arquivos devem ser passados por parâmetro na chamada do executável:

`./tp2[o,h] input.txt output.txt`

O arquivo de entrada possui um inteiro N na primeira linha onde N é o número de instâncias a serem simuladas. Em seguida, as N instâncias são definidas da seguinte forma. A primeira linha possui um inteiro M indicando o número de *ERBs* existentes na cidade (para solução ótima: $1 \leq M \leq 400$, para a heurística: $1 \leq M \leq 4.000$). As linhas seguintes serão listadas da seguinte forma ($X \Rightarrow 0, Y \geq 0, R \geq 0$), onde X e Y são as coordenadas e R o raio de cada antena. As *ERBs* serão representadas por pares de inteiros distintos (x,y) representando as coordenadas e um raio (r). Considere que não há o caso em que toda área de atuação de uma antena seja englobada por uma outra (i.e. queremos evitar *ERBs* redundantes). Lembre-se que circunferência, gerada pela atuação da antena, também é considerada dentro da área de atuação (i.e. Circunferências que

se tocam são passíveis de interferência).

Para cada instância, deve ser impresso no arquivo de saída o número canais de utilizados para ativar todas as *ERBs* de forma que não haja possibilidade de interferência.

Exemplo

A seguir temos um exemplo de funcionamento do programa para a solução ótima:

Entrada:

```
1
8
17 36 7
11 26 6
25 28 6
19 22 6
14 14 5
26 14 5
42 42 5
43 32 6
```

Saída:

```
2
```

A figura 3 ilustra o exemplo, mostrando o grafo gerado.

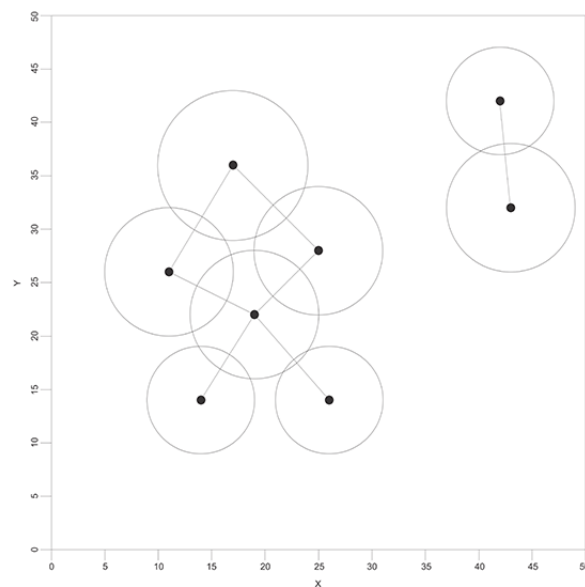


Figura 3: Grafo gerado pelo arquivo de entrada dado como exemplo

Entrega

- A data de entrega desse trabalho é **13 de Maio**.
- A penalização por atraso obedece à seguinte fórmula $2^{d-1}/0.32\%$, onde d são os dias úteis de atraso.
- Submeta apenas um arquivo chamado `<numero_matricula>_<nome>.zip`. Não utilize espaços no nome do arquivo. Ao invés disso utilize o caractere '_'.
- Não inclua arquivos compilados ou gerados por IDEs. **Apenas** os arquivos abaixo devem estar presentes no arquivo zip.
 - Makefile
 - Arquivos fonte (*.c e *.h)
 - Documentacao.pdf
- Não inclua **nenhuma pasta**. Coloque todos os arquivos na raiz do zip.
- Siga rigorosamente o formato do arquivo de saída descrito na especificação. Tome cuidado com whitespaces e formatação dos dados de saída
- **NÃO SERÁ NECESSÁRIO ENTREGAR DOCUMENTAÇÃO IMPRESSA!**
- Será adotada **média harmônica** entre as notas da **documentação e da execução**, o que implica que a nota final será 0 se uma das partes não for apresentada.

Documentação

A documentação não deve exceder 10 páginas e deve conter pelo menos os seguintes itens:

- Uma **introdução** do problema em questão.
- **Modelagem e solução proposta** para o problema. O algoritmo deve ser explicado de forma clara, possivelmente através de pseudo-código e esquemas ilustrativos.
- **Análise de complexidade** de tempo e espaço da solução implementada.
- **Experimentos** variando-se o tamanho da entrada e quaisquer outros parâmetros que afetem significativamente a execução.
- Especificação da(s) **máquina(s) utilizada(s)** nos experimentos realizados.
- Uma breve **conclusão** do trabalho implementado.
- Espera-se gráficos que explorem o tamanho da entrada do problema (número de ERBs) versus tempo de execução e comparem a qualidade do algoritmo aproximado com a busca exata.
- **Ponto Extra:** Demonstrar a razão de aproximação da heurística implementada.

Código

- O código deve ser obrigatoriamente escrito na **linguagem C**. Ele deve compilar e executar corretamente nas máquinas Linux dos laboratórios de graduação.
- O utilitário ***make*** deve ser utilizado para auxiliar a compilação, um arquivo *Makefile* deve portanto ser incluído no código submetido.
- As estruturas de dados devem ser **alocadas dinamicamente** e o código deve ser **modularizado** (divisão em múltiplos arquivos fonte e uso de arquivos cabeçalho .h)
- **Variáveis globais** devem ser evitadas.
- Parte da correção poderá ser feita de forma automatizada, portanto **siga rigorosamente os padrões de saída especificados**, caso contrário sua nota pode ser prejudicada.
- **Legibilidade e boas práticas** de programação serão avaliadas.

Log alterações

: 30-04 Alteração dos valores do exemplo e da Figura 3.