

Projeto e Análise de Algoritmos II
Trabalho Prático
Prof. Eduardo C. Xavier

Você possui uma empresa reconhecida na área de otimização e algoritmos. Um dia é chamado por telefone para uma conversa urgente com um grupo de pessoas que está trabalhando em um grande projeto: “Salvar o mundo”. Eles lhe informam que trabalham para um ministério da ONU sob supervisão de vários países. Devido a dificuldade para se estabelecer um acordo sobre metas de redução de emissão de carbono por parte dos países, este grupo deve achar uma outra solução para o problema do aquecimento global.

Este grupo é formado por cientistas e engenheiros, que desenvolveram uma solução (não muito boa mas a melhor possível): Emitir dióxido de enxofre na estratosfera, pois este gás reflete grande parte dos raios solares e funcionaria como um filtro solar para a Terra, invertendo a trajetória de aumento de temperatura. A idéia é construir torres com canos enormes que levariam o dióxido de enxofre até a estratosfera. Estes são chamados de sistemas de emissão. Dependendo do lugar onde este gás é liberado, ele cobrirá diferentes partes do céu, devido as correntes de ar existentes. Dado uma posição para instalação do sistema de emissão, e dado as correntes de ar existentes, pode-se prever com certa precisão quais lugares da Terra será coberto por esta camada “protetora”. Para a instalação de um sistema de emissão, há um custo monetário e ecológico que varia dependendo do lugar onde este é instalado. O objetivo é determinar os lugares de instalação dos sistemas de emissão de tal forma que toda a Terra seja coberta pelo dióxido de enxofre, e ao mesmo tempo se minimize o custo de instalação dos sistemas de emissão. O espaço contínuo a ser coberto (a superfície da Terra) será discretizado, de tal forma que temos pontos que devem ser cobertos (veja Figura).

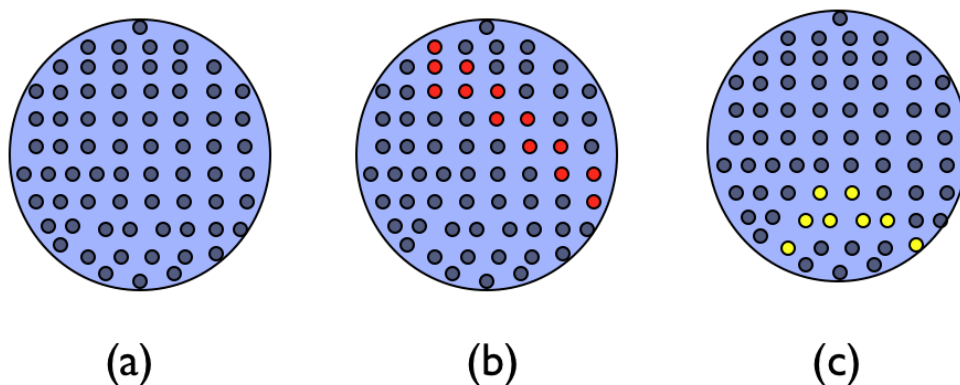


Figura 1: (a) Exemplo de Terra com superfície discretizada (b) Pontos cobertos por um sistema de emissão (c) Pontos cobertos por um segundo sistema de emissão.

Você diz para o grupo que vai pensar sobre o problema. Mais tarde no seu escritório, após algumas tentativas de projeto de algoritmos polinomiais você começa a desconfiar que o problema é NP-Difícil. Você começa a perceber que este problema é parecido com o SetCover (de fato você pode reduzir o SetCover para este problema). Você consegue mostrar que o problema é NP-Difícil! Você propõe então a implementação de uma heurística para o problema.

Neste trabalho vocês devem implementar uma heurística para o problema descrito anteriormente. A heurística

implementada deve ser uma dentre: Busca Tabu, GRASP, Algoritmos Genéticos e Simulated Annealing. Alguns detalhes sobre o trabalho:

- O trabalho deve ser feito em grupos de 2 ou 3 pessoas.
- A sua heurística deve ser implementada em C, C++ ou Java padrões (C/C++ compilável com gcc, e Java com java da SUN/ORACLE).
- Vocês devem **entregar via email até o dia 12 de junho** um único arquivo Gi.zip, onde Gi é o nome atribuído ao seu grupo (disponível na página da disciplina). Ao descompactar o arquivo deve ser criado um diretório chamado Gi contendo:
 1. Diretório chamado **código**: contém todos os arquivos da sua implementação bem como um arquivo makefile para compilar o código. O arquivo executável gerado deve ter o nome saveWorld .
 2. Arquivo pdf que é um relatório do trabalho: Vocês devem escrever um relatório de no máximo 10 páginas, contendo pelo menos: 1) Os nomes e RAs dos integrantes do grupo 2) Organização e breve descrição dos arquivos código fonte 3) Heurística utilizada 4) Descrição em alto nível da sua implementação mostrando o funcionamento geral da heurística; quais estruturas de dados foram usadas (para representar soluções para acelerar geração de vizinhança etc); quais operadores foram usados; qual a vizinhança utilizada; descrição de setup de parâmetros da heurística; etc 5) Uma seção com resultados computacionais descrevendo testes realizados, soluções encontradas e tempo de execução (use tabelas e gráficos para mostrar os resultados).
 3. No email enviado para o professor deve constar no *subject* apenas o texto: **trabalho mc548**.
 4. A heurística deve rodar por no máximo 1 minuto, e deve então ser impressa uma solução do problema (mais detalhes abaixo).

No endereço <http://www.ic.unicamp.br/~eduardo/mc548-2011-1/geradorInst.py> vocês encontram um exemplo de *script* python para gerar instâncias disponibilizado pelo professor. As instâncias são geradas de forma aleatória. No relatório vocês devem reportar testes com instâncias geradas com este *script*. Vocês podem também gerar outras instâncias, e neste caso vocês devem descrever detalhes sobre a geração de tais instâncias. Reportem número de pontos a serem cobertos, número de subconjuntos, como foram criados os subconjuntos etc.

Cada arquivo de entrada tem o seguinte formato:

```
N Número de pontos a serem cobertos
M Número de estações de emissão
S_1 C_1 lista de pontos cobertos por estação S_1
S_2 C_2 lista de pontos cobertos por estação S_2
...
```

A primeira linha indica quantos pontos devem ser cobertos (simplificamos os pontos simplesmente como números de $1, \dots, N$). Na linha seguinte é especificado o número de estações de emissão. Nas linhas seguintes especificamos as estações com nome S_i , um custo real dado por C_i e em seguida uma lista de números separados por espaço. Os números estão entre 1 e N , e correspondem a pontos cobertos caso seja instalado a estação S_i .

Objetivo: Selecionar estações que cobrem todos os pontos, tal que a soma dos custos das estações seja a menor possível.

Exemplo de arquivo de entrada:

```
N 10
M 4
S_1 34.87 1 4 10 3
S_2 140.8 2 3 5 6 8 9
S_3 100.8 5 6 10 3 6
S_4 50.68 1 2 3 7
```

O seu programa deve aceitar um arquivo de entrada pela linha de comando (`saveWorld inst`) e deve gerar uma saída da seguinte forma:

```
Valor: x
Total: n
S_i1
....
S_in
```

Na primeira linha deve ser impresso "Valor: x" onde x corresponde ao valor da solução (soma dos custos das estações usadas). Na linha seguinte deve ser impresso "Total: n" onde n deve corresponder ao número de estações da sua solução. Nas linhas seguintes devem ser impressos os nomes dos n conjuntos utilizados, um conjunto por linha. **Siga corretamente estes modelos de impressão de solução. Preste atenção no ":" e no espaço em branco para depois imprimir x e n.**

Saída Exemplo:

```
Valor: 226.35
Total: 3
S_1
S_2
S_4
```

Importante: O seu programa deve executar no máximo 1 minuto para cada instância, e deve então imprimir a melhor solução encontrada.

Plágio ou cópias de trabalho não serão aceitas! Vocês podem porém usar idéias descritas em artigos que apresentam heurísticas para o problema SetCover. Neste caso vocês devem fazer a própria implementação da heurística, e devem citar os artigos de onde tiraram idéias de implementação.

Os trabalhos que não satisfaçam as restrições descritas neste texto terão nota 0.

Sobre a nota: A nota de um trabalho entregue varia de 0 a 10, mas 2 pontos serão atribuídos comparando-se os resultados obtidos entre os trabalhos entregues pelos alunos. O professor realizará testes de comparação entre as heurísticas e as melhores receberão as melhores notas.