# Lista de exercícios 2

**Exercício 1 (Algoritmos gulosos, 2pt)**
Quais dos seguintes sistemas de conjuntos é independente? Quais satisfazem a propriedade de troca? Prova as afirmações.

a) Dado um conjunto de intervalos $I$, cada um com um peso $w_i$, $i \in I$, queremos encontrar o conjunto de intervalos de maior peso tal que nenhum par de intervalos no conjunto selecionado possui uma interseção. O sistema de conjuntos correspondente consiste do universo $I$ e todos subconjuntos de $I$ sem interseção.

b) Dado um grafo não direcionado $G = (V, A)$, com pesos $w_a$, $a \in A$ nas arestas queremos encontrar a partição $V_1 \dot\cup V_2 = V$ que maximiza o valor do corte entre os partes. O valor do corte é o peso total das arestas entre $V_1$ e $V_2$. O sistema de conjuntos correspondentes consiste do universo $U = V$ e de todos subconjuntos de $U$.

**Exercício 2 (Busca Tabu para QBP, 3pt)**

a) Implemente e avalia uma busca tabu para o QBP. Usa a vizinhança 1-flip da lista 1. Após da trocar o valor uma variável é declarada tabu por $d$ iterações (duração tabu). A busca tabu termina depois de $20n$ iterações sem melhorar a melhor solução conhecida.

b) Avalia o algoritmo com durações tabu $d \in \{1/25, 1/50, 1/100, 1/200\}n$ para $n$ variáveis. Avalia 10 replicações em cada instância. Usa as instâncias da lista 1 e compare com os resultados das buscas locais.

c) Repete o experimento do item anterior, com duração tabu $d \in \{1/25, 1/50, 1/100, 1/200\}n + r$ com um número aleatório $r \in [1, 10]$. Compare.

**Exercício 3 (Heurística construtiva para QBP, 2pt)**

a) Implemente um algoritmo guloso randomizado para o QBP. Inicialmente define todas variáveis como “livres”, i.e. sem valor. Repetidamente, atribui um valor 1 à variável livre que diminui a função objetivo o mais possível. O valor da função objetivo é calculado supondo que as variáveis livres possuem valor 0. Para randomizar o algoritmo, use a estratégia guloso-$\alpha$.

b) Testa a sua construção com valores $\alpha \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ e com 1000 replicações para cada instância da lista 1. Plote seis histogramas que mostram a distribuição das soluções gulosas para os diferentes valores de $\alpha$.

c) (Opcional, 2pt) Estende a heurística construtiva da questão anterior para um otimização de colônias de formigas. Compare com a heurística construtiva acima e a busca local da primeira lista.

**Exercício 4 (Busca dispersa para QBP, 3pt)**
Neste exercício vamos estudar a busca dispersa aplicada ao problema da programação quadrática binária.

**Preparação do código** Nas notas de aula, no Algoritmo 4.2, temos o pseudocódigo para a busca dispersa. Segue as seguintes decisões de projeto (as linhas são referentes ao pseudocódigo nas notas de aula):

- Linha 2: Crie o conjunto inicial de soluções diversas sorteando $|C|$ soluções aleatórias.
- Linhas 3 e 11: Utilize a técnica de seleção de soluções de referência a partir de uma população indicada nas notas para criar e atualizar o conjunto de referência.
- Linha 5: Gere a família de subconjuntos selecionando todos os pares de elementos do conjunto de referência.
- Linha 8: Recombine cada par de soluções via religamento de caminhos.

- Linha 9: Utilize uma busca local monótona como processo de melhoria de novas soluções. Você pode reutilizar a busca local da lista anterior.

Utilize a distância de Hamming como métrica de distância entre as duas soluções para construir a vizinhança reduzida e para construção do conjunto de referência. Durante a busca local e o religamento de caminhos utilize a técnica "first improvement".

**Estudo experimental** Utilize as oito instâncias bqp{50,100,250,500}-$x$, com $x \in \{1, 7\}$, disponíveis na biblioteca Biq Mac (http://biqmac.uni-klu.ac.at/biqmaclib.html). Os valores das soluções ótimas estão disponíveis em http://biqmac.uni-klu.ac.at/biqmaclib.pdf.
Como medida da qualidade da solução encontrada utilize o desvio percentual da melhor solução conhecida $d = |\frac{s-o}{o}|$ onde $s$ é o valor da solução encontrada e $o$ da melhor solução conhecida.

a) Fixe o parâmetro $|R| = 20$ (que implica $|C| = 190$) e utilize a estratégia de religamento de caminhos para frente. Execute o algoritmo até encontrar a solução de melhor valor conhecido, ou até um tempo limite de 5 min para cada instância e cada par de $b_1 = 2y$ e $b_2 = |R| - 2y$, com $y \in [1, 9]$. Apresente uma tabela com a qualidade média do resultado, o número médio de iterações e o tempo médio de execução para cada configuração de $b_1$ e $b_2$. Quais configurações de parâmetros encontram as soluções mais rápido, ou caso não encontram geram os melhores resultados?

b) Utilizando a melhor configuração de parâmetros teste a influência da busca local: Execute o algoritmo com o mesmo critério de parada para cada instância, uma vez com busca local habilitada e uma vez desabilitada. Apresente uma tabela demonstrando a influência da busca local. Inclua a qualidade média da solução, o número médio de iterações e o tempo médio. Os dados indicam que deve manter a etapa da busca local na heurística?

c) (Opcional, 1pt) Utilizando a melhor configuração de parâmetros teste a influência da estratégia de religamento de caminhos. Execute o algoritmo com o mesmo critério de parada para cada instância uma vez com o religamento de caminhos para frente outra vez com religamento de caminhos para trás. Apresente uma tabela comparando os resultados. Os dados indicam alguma diferença no comportamento do algoritmo?

**Data de entrega**: 07/05/2014.
Observações:

- Para plotar um histograma (sugestão): gera um arquivo `x.dat` com um valor por linha. Chama "R" (GNU R) e usa os comandos

  ```
  d<-read.table("x.dat")
  pdf("x.pdf")
  hist(d$V1)
  dev.off()
  q()
  ```

  para produzir o histograma em `x.pdf`.