

Lista de exercícios 1

Exercício 1 (Vizinhanças no PCV, 1.5pt)

Considere a seguinte vizinhança do PCV. Um vizinho de uma rota é obtido por remover uma cidade e inseri-la em qualquer outra posição da rota novamente. Essa vizinhança é simétrica? Fracamente otimamente conectada? Exata?

Exercício 2 (Aproximação do PCV por busca local, 2.5pt)

Mostra que nenhuma busca local para o PCV com uma vizinhança que é analisável em tempo polinomial consegue uma garantia de aproximação melhor que $\alpha(n)$, para qualquer função $\alpha(n)$ computável em tempo polinomial, caso $P \neq NP$.

Exercício 3 (Busca local para $Pm \parallel C_{\max}$, 2.5pt)

Considere a seguinte vizinhança do problema $Pm \parallel C_{\max}$ (dado n tarefas com tempo de processamento p_i , $i \in [n]$ e m máquinas distribui as tarefas de forma que o tempo de término da última tarefa C_{\max} é minimizada). Dado uma solução, um vizinho é obtido por mover uma tarefa de uma máquina para uma outra máquina.

Uma máquina com tempo de execução C_{\max} se chama *crítica*. Observe que para reduzir C_{\max} é necessário mover uma tarefa de uma máquina crítica, i.e. a vizinhança pode ser reduzida. Caso existe mais que uma máquina com tempo de execução C_{\max} o valor não vai diminuir. Por isso vamos usar a função objetivo (C_{\max}, c) com c o número de máquinas críticas. Um par (C, c) é considerado menor que (C', c') caso $C < C'$ ou $C = C'$ e $c < c'$.

- Implementa essa busca local em instâncias com $m = 10, 20, 50$ e $n = m^r$ tarefas com $r = 1.2, 1.5, 2$ e tempos de processamento aleatórios em $[1, 100]$. Aplique a busca local para cada combinação de parâmetros em 10 instâncias geradas aleatoriamente. Compare as estratégias “primeira melhora” e “melhor melhora” em tempo e número de iterações. Determine a complexidade empírica do algoritmo.
- Compare com a solução obtida por processar as tarefas na ordem dada, sempre alocando a tarefa atual na máquina menos ocupada. (Este algoritmo é uma $2 - 1/m$ -aproximação.)
- (Opcional): Mostra que a busca local acima é uma $2 - 2/(m + 1)$ -aproximação (2pt) ou uma 2-aproximação (1pt).

Entrega dos resultados:

- Um arquivo texto com os resultados individuais, da forma

```
alg n   m   rep time iterations value
PM 16  10   1   3.3  1234         88123
MM 16  10   1   2.5  5678         88124
LS 16  10   1   2.5   NA         88124
...
```

com PM os resultados da “primeira melhora”, MM da “melhor melhora” e LS (“list scheduling”) do item b. Observe que LS não é um método iterativo, logo a coluna “iterations” informa o valor “NA”. A coluna “rep” é o número da repetição com uma instância diferente.

Exercício 4 (Busca local para QBP, 3.5pt)

Estuda a busca local no problema de programação quadrática binária (ingl. quadratic binary programming, QBP). Usa as instâncias `bqp{50,100,250,500}-1` disponíveis na biblioteca Biq Mac (<http://biqmac.uni-klu.ac.at/biqmaclib.html>). Os valores das soluções ótimas são disponíveis em <http://biqmac.uni-klu.ac.at/biqmaclib.pdf>.

- Compare as estratégias “first improvement” e “best improvement”.

- b) Compare com uma busca local monótona randomizada. Testa valores de $p \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$. Repete o experimento para cada valor de p cinco vezes. Relata o tempo até encontrar a solução ótima, ou o melhor valor encontrada depois de um tempo limite de 5 min e o número de iterações numa tabela da forma $\mu \pm \sigma$.
- c) Determine a complexidade empírica do algoritmo (o tempo para encontrar a solução ótima em função do tamanho da instância).

Observações:

- Para um desempenho razoável é necessário implementar um cálculo eficiente da função objetivo conforma a observação 2.1 nas notas.
- Para escolher um vizinho melhor ou arbitrário uniformemente sem armazenar todos vizinhos pode-se usar *reservoir sampling*: ao encontrar o i -ésimo vizinho candidato, aceitá-lo com probabilidade $1/i$.

Entrega dos resultados:

- Para o primeiro item um arquivo texto com os resultados individuais da forma

```
alg instance      time iterations      value
PM  bqp50         17.3 23764          88123
MM  bqp50         32.2 41234          88123
...
```

com o tempo em segundos.

- Para o segundo item um arquivo texto com os resultados individuais, da forma

```
p  instance      rep time iterations value
0.0 bqp50         1  30.1 2943          88123
0.0 bqp50         2  33.8 1234          88124
...
```

com o tempo em segundos. A coluna “rep” é o número de repetição do algoritmo na mesma instância.

Data de entrega: 11/04/2014.

Regras para listas de exercícios

1. Os exercícios podem ser resolvidos em colaboração com outros, mas a entrega é individual informando os eventuais colaboradores.
2. A entrega consiste das respostas (formato PDF, não escrito a mão), o código para questões de implementação e os resultados experimentais detalhadas, caso tem experimentos.
3. Para receber pontos as respostas devem ser justificadas (i.e. provadas quando não são óbvias).
4. Somente entregam respostas que vocês sabem explicar pessoalmente.