

comentarios-dataset-problema-da-mochila

Contextualização

O dataset (conjunto de instâncias) para o problema da mochila foram retirados do artigo

Pisinger, D., **Where are the hard knapsack problems?** *Computers & Operations Research*, 2005. 32(9): p. 2271-2284"

Essas instâncias estão armazenadas [neste site](#) e também encontram-se no [site da disciplina](#).

Para este projeto, estamos interessados nas instâncias classificadas como large-scale, elas estão no diretório `large_scale`.

Neste diretório, existem três tipos de instâncias, com diferentes tipos de correlação entre os dados de peso e valor dos itens.

1. Arquivos `knaPI_1` : correlacionam fracamente os valores de peso e valor dos itens.
2. Arquivos `knaPI_2` : correlacionam de maneira moderada os valores de peso e valor dos itens.
3. Arquivos `knaPI_3` : correlacionam de maneira forte os valores de peso e valor dos itens.

A correlação tem a ver com a dependência dos valores das duas variáveis. Isto é, para uma correlação forte, os valores de peso e valor tendem a serem mais próximos, enquanto em uma correlação mais fraco, os itens possuem mais disparidade entre o peso e o seu valor.

A discussão deverá ser realizada de acordo com as diferentes correlações dos valores, visto que elas possuem influência direta nos resultados.

Sugestão de configuração experimental

É interessante comparar as duas abordagens gulosas e a abordagem de programação dinâmica de acordo com os três tipos de arquivos: `knaPI_1`, `knaPI_2` e `knaPI_3`. Tanto o tempo de execução quanto a qualidade da solução devem ser avaliadas.

Descrição dos arquivos

Os arquivos estão nomeados da seguinte forma

`knaPI_x_y_z_1` em que:

- `x` corresponde ao grau de correlação entre peso e valor dos itens:
 - `x=1` para baixa correlação.
 - `x=2` para média correlação.
 - `x=3` para alta correlação.
- `y` corresponde à quantidade de itens.
- `z` corresponde ao peso máximo de cada item. No caso desse dataset, `z` é sempre 1000.

A primeira linha de cada arquivo possui dois inteiros:

- y , conforme explicado acima e W , correspondente à capacidade da mochila.
As próximas y linhas de cada arquivo possuem cada, dois valores v_i e w_i , que descrevem, respectivamente, o valor de o peso de cada um dos itens.

Por fim, a última linha do arquivo possui um vetor X , de tamanho y contendo apenas os valores 0 e 1. Se $X[i] = 0$ então, o i -ésimo item, que possui valor v_i e peso w_i **não** está na melhor solução possível. Se $X[i] = 1$, então o i -ésimo item, está no melhor valor possível. O vetor X pode ser utilizado para descobrir o valor máximo do benefício que pode ser acumulado e comparar com a implementação do algoritmo de programação dinâmica, que deve chegar no mesmo valor. Como é possível que haja mais de um arranjo de itens que chegue na solução ótima, o vetor X não é único! A solução do algoritmo de programação dinâmica implementado por vocês pode escolher itens diferentes e chegar na solução ótima também.

Métrica da qualidade da solução

Para avaliar a qualidade da solução, pode-se utilizar a seguinte métrica:

$$q = \frac{x}{x^*}$$

Onde x corresponde ao benefício máximo encontrado pela solução enquanto x^* corresponde ao benefício ótimo, isto é, à melhor solução possível. Quanto mais próximo de 1 for esse valor, melhor. Soluções que sempre encontram a melhor resposta, tem valor $q = 1$ para qualquer instância.

Gráficos

A sugestão é fazer gráfico separando os arquivos de acordo com o nível de correlação de peso e valor dos itens. Ou seja, haverá gráficos para os arquivos `knaPI_1`, `knaPI_2` e `knaPI_3`.

Para cada um desses tipos, haverá dois gráficos:

- Um gráfico de curvas para o tempo de execução, conforme y aumenta. Neste caso, serão 3 curvas, um para cada abordagem. De acordo com o dataset, y pode assumir os valores (100, 200, 500, 1000, 2000, 5000, 1000), logo, são sete pontos para cada curva.
- Um gráfico de barras com a qualidade da solução, conforme y aumenta. No caso, são 3 barras para cada valor de y , uma para cada tipo de abordagem. De acordo com o dataset, y pode assumir os valores (100, 200, 500, 1000, 2000, 5000, 1000), logo, são sete grupos de três barras, totalizando vinte e uma barras.