

Proposta Inicial Para o Trabalho Final da Disciplina de Otimização Combinatória

Bruno Santana Massena de Lima

Novembro 2018

1 Problema da Mochila Conexa

Formulação:

$$\begin{aligned} \max \quad & \sum_{i \in [V]} v_i c_i \\ \text{s.t.} \quad & \sum_{i \in [V]} c_i w_i \leq U \\ & f_{ij} \leq M E_{ij}, \quad \forall i, j \in [V] \\ & f_{ij} \leq M c_i, \quad \forall i, j \in [V] \\ & f_{ij} \leq M c_j, \quad \forall i, j \in [V] \\ & \sum_{j \in [V]} f_{ij} + 1 \leq \sum_{j \in [V]} f_{ji} + M(s_i + (1 - c_i)), \quad \forall i \in [V] \\ & \sum_{j \in [V]} f_{ij} + 1 \geq \sum_{j \in [V]} f_{ji} - M(s_i + (1 - c_i)), \quad \forall i \in [V] \\ & \sum_{j \in [V]} f_{ij} \geq \sum_{i \in [V]} c_i - 1 - M(1 - s_i), \quad \forall i \in [V] \\ & \sum_{i \in [V]} f_{ij} \leq M(1 - s_j), \quad \forall j \in [V] \\ & \sum_{i \in [V]} s_i = 1 \\ & 0 \leq c_i \leq 1, \quad \forall i \in [V] \\ & 0 \leq s_i \leq 1, \quad \forall i \in [V] \\ & f_{ij} \geq 0, \quad \forall i, j \in [V] \end{aligned}$$

Nesta formulação:

E representa a matriz de adjacência do grafo de entrada

V representa a quantidade de vértices do grafo de entrada

U representa a capacidade máxima da mochila

v_i representa o valor do vértice i

w_i representa o peso do vértice i

c_i indica se o vértice i foi escolhido (1) ou não (0)

s_i indica se o vértice i é o vértice de origem do fluxo (1) ou não (0)

f_{ij} representa o fluxo do vértice i ao vértice j

2 Elementos do Simulated Annealing

A **representação de uma solução** se dará por uma lista de inteiros, cada um representando um item/vértice escolhido pertencente à instância do problema.

A **solução inicial** será feita de forma aleatória. O algoritmo irá selecionar um vértice inicial com peso menor que a capacidade máxima da mochila. A partir disso, o algoritmo irá encontrar vértices adjacentes aos já adicionados na solução e adicioná-los à mesma. Este processo se repetirá até que não seja possível adicionar mais vértices à solução sem que o peso total dos vértices ultrapasse a capacidade máxima da mochila. Além disso, também penso em implementar outros métodos de escolha de solução inicial, algo como uma escolha gulosa, ou *best-fit/worst-fit*

Vizinhanças: Inicialmente, a vizinhança será aleatória. Será selecionado um item da solução atual para ser removido e um item para ser adicionado. Caso a solução gerada seja factível, ela será retornada.

Caso contrário, o processo será repetido. Para diversificar a quantidade de itens na mochila, também pode ocorrer de algum dos vértices não ser escolhido, ou seja, um item será adicionado sem a necessidade de remover outro, ou vice-versa. Além desta, também pretendo implementar outros tipos de vizinhanças, e discutir qual obteve melhores resultados no relatório final.

Temperaturas: Para definir as temperaturas inicial e final, usarei os métodos descritos nas notas de aula:

”Exemplo 10.2 (Temperatura inicial) Defina uma probabilidade p_i Execute uma versão rápida (I pequeno) do algoritmo para determinar uma temperatura inicial tal que um movimento é aceito com probabilidade p_i

Exemplo 10.3 (Temperatura final) Defina uma probabilidade p_f Para cada nível de temperatura em que os movimentos foram aceitos com probabilidade menos que p_f incrementa um contador. Zera o contador caso uma nova melhor solução é encontrada. Caso o contador chega em 5, termina.”