

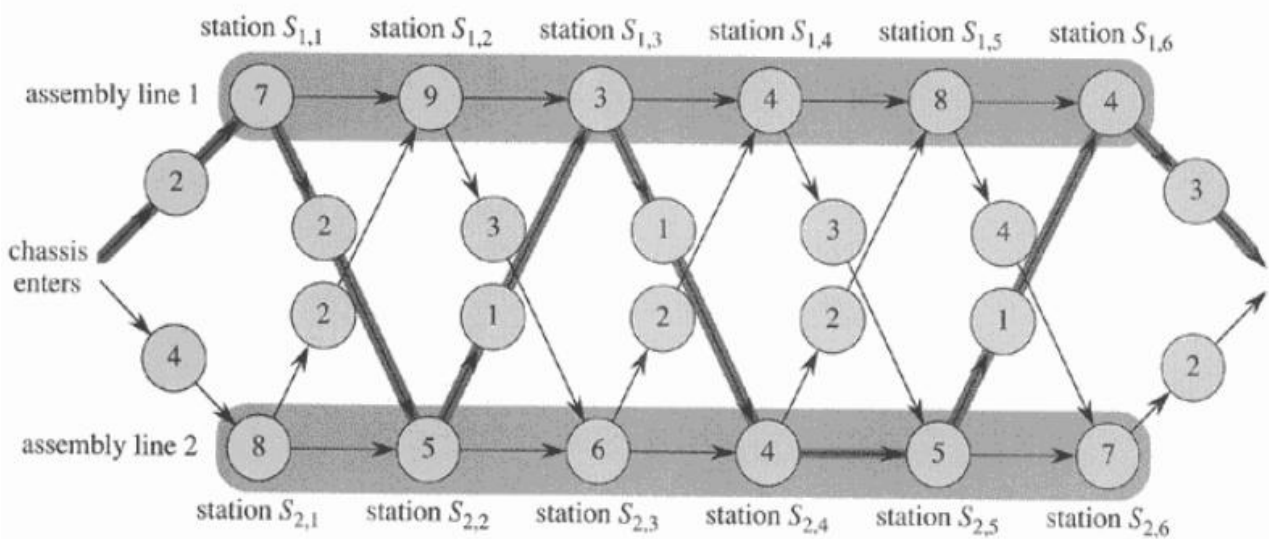
Linha de Montagem em Java

Gustavo Ceconelli,
Mateus Mourão,
Tiago Araújo.

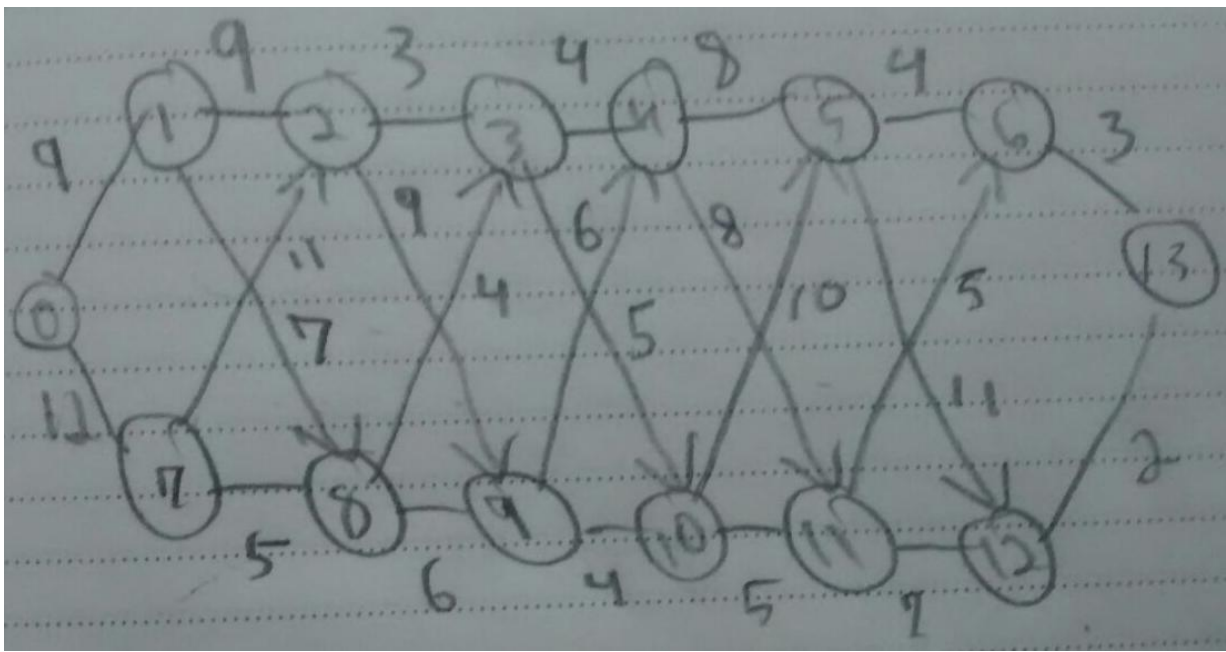
Novembro de 2017

Programação gulosa

Na programação gulosa foi utilizado o algoritmo de Dijkstra, que vai adicionando vértices ao caminho de acordo com a aresta de menor peso que leva a um novo vértice. Isso ocorre de acordo com os caminhos mínimos que já conhecemos dos vértices previamente descobertos. Para representar a linha de montagem em um grafo, foi considerado que o estágio inicial e final são vértices. Como mostrado à seguir:



A linha de montagem acima é representada pelo grafo abaixo, onde devemos achar o caminho mínimo de 0 a 13:



Programação dinâmica

Com programação dinâmica temos as informações armazenadas em vetores, e também representadas pelas linhas. Para encontrar a solução do problema, utilizamos soluções menores armazenadas de subproblemas. Para encontrar o resultado, calculamos o menor caminho até a estação 1 (tanto da linha 1 quanto da 2), depois calculamos o menor caminho até a estação 2 (da linha 1 e 2) com base nos caminhos já encontrado até as duas estações 1, e assim por diante.

Resultados

O caminho mínimo encontrado para:

- a. A1 = [02, 07, 09, 03, 04, 08, 04, 03]
- b. A2 = [04, 08, 05, 06, 04, 05, 07, 02]
- c. T1 = [02, 03, 01, 03, 04]
- d. T2 = [02, 01, 02, 02, 01]

Pelo algoritmo guloso:

Aresta 0 a 1; Tempo-> 9

Aresta 1 a 8; Tempo-> 7

Aresta 8 a 3; Tempo-> 4

Aresta 3 a 10; Tempo-> 5

Aresta 10 a 11; Tempo-> 5

Aresta 11 a 6; Tempo-> 5

Aresta 6 a 13; Tempo-> 3

Tempo total: 38

Pelo algoritmo dinâmico:

(de trás para frente)

Linha: 1 Estação: 6

Linha: 2 Estação: 5

Linha: 2 Estação: 4

Linha: 1 Estação: 3

Linha: 2 Estação: 2

Linha: 1 Estação: 1

Custo total: 38

O resultados resultados obtidos tem o mesmo tempo total, mas realizam etapas diferentes, pois trabalham o problema de forma diferente, mas encontram o menor caminho da mesma forma

O caminho mínimo encontrado para:

a. A1 = [03, 05, 07, 10, 05, 09, 11, 09, 05, 02, 06]

b. A2 = [02, 06, 03, 09, 11, 04, 09, 03, 12, 04, 05]

c. T1 = [03, 05, 04, 02, 07, 05, 08, 01]

d. T2 = [05, 03, 07, 05, 06, 02, 05, 02]

Pelo algoritmo guloso:

Aresta 0 a 10; Tempo-> 8

Aresta 10 a 11; Tempo-> 3

Aresta 11 a 3; Tempo-> 13

Aresta 3 a 4; Tempo-> 5

Aresta 4 a 14; Tempo-> 6

Aresta 14 a 15; Tempo-> 9

Aresta 15 a 16; Tempo-> 3

Aresta 16 a 8; Tempo-> 10

Aresta 8 a 9; Tempo-> 2

Aresta 9 a 19; Tempo-> 6

Tempo total: 65

Pelo algoritmo dinâmico:

(de trás para frente)

Linha: 1 Estação: 9

Linha: 1 Estação: 8

Linha: 2 Estação: 7

Linha: 2 Estação: 6

Linha: 2 Estação: 5

Linha: 2 Estação: 4

Linha: 2 Estação: 3

Linha: 2 Estação: 2

Linha: 2 Estação: 1

Custo total: 65

O resultados resultados obtidos tem o mesmo tempo total, mas descrevem etapas diferentes, pois trabalham o problema de forma diferente, mas encontram o menor caminho da mesma forma

O caminho mínimo encontrado para:

e. A1 = [05, 10, 06, 03, 08, 05, 03, 07, 12, 08]

f. A2 = [07, 03, 05, 03, 07, 06, 04, 09, 10, 09]

g. T1 = [04, 02, 07, 02, 05, 08, 02]

h. T2 = [06, 01, 07, 03, 06, 04, 05]

Pelo algoritmo guloso:

Aresta 0 a 9; Tempo-> 10

Aresta 9 a 10; Tempo-> 5

Aresta 10 a 3; Tempo-> 4

Aresta 3 a 4; Tempo-> 8

Aresta 4 a 5; Tempo-> 5

Aresta 5 a 6; Tempo-> 3

Aresta 6 a 7; Tempo-> 7

Aresta 7 a 8; Tempo-> 12

Aresta 8 a 17; Tempo-> 8

Tempo total: 62

Pelo algoritmo dinâmico:

(de trás para frente)

Linha: 1 Estação: 8

Linha: 1 Estação: 7

Linha: 1 Estação: 6

Linha: 1 Estação: 5

Linha: 1 Estação: 4

Linha: 1 Estação: 3

Linha: 2 Estação: 2

Linha: 2 Estação: 1

Custo total: 62

O resultados resultados obtidos tem o mesmo tempo total, e descrevem caminhos iguais.