Otimização de Sistemas

Algoritmos Construtivos

Conteúdo

- Métodos Construtivos
  - Problema de Roteamento de Veículos
    - Algoritmo de Clarke-Wright para o VRP: Economias/ Savings

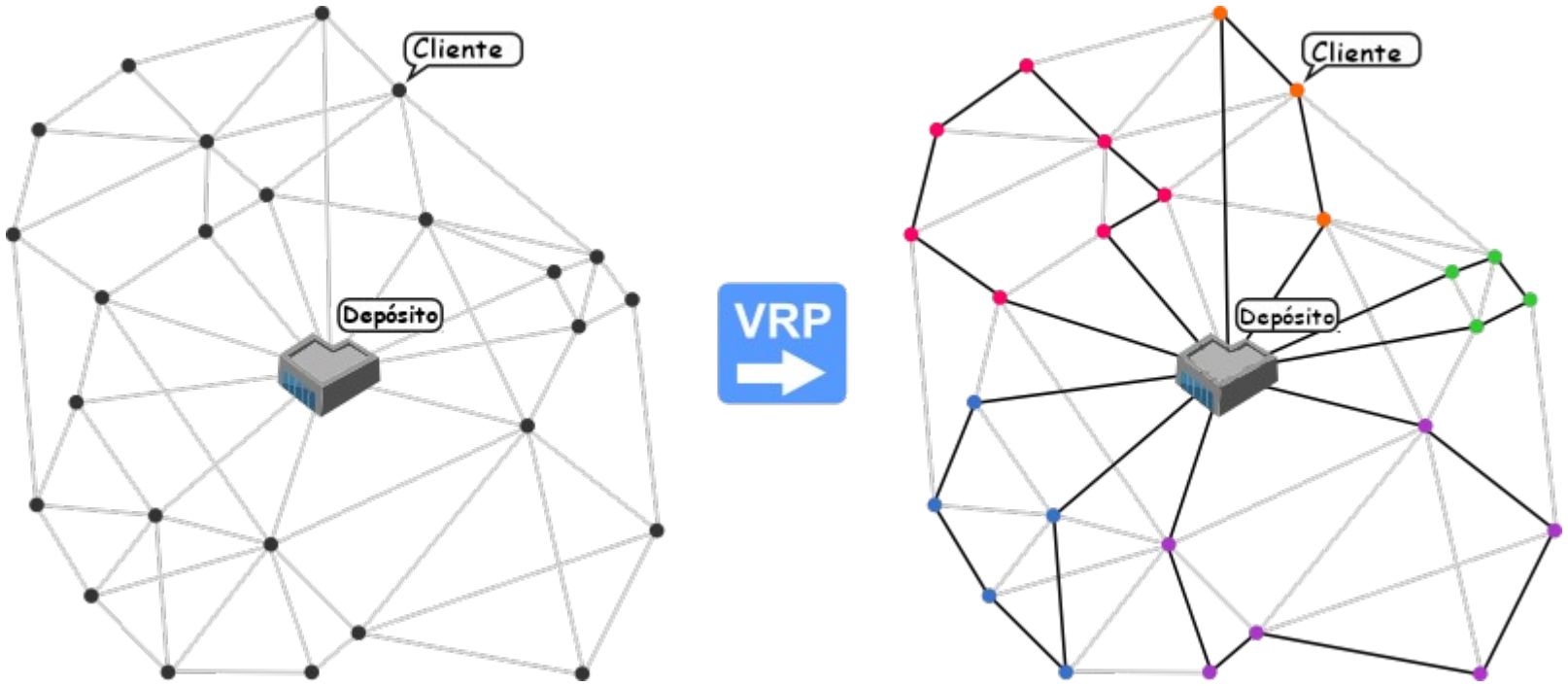
Problema Roteamento de Veículos

- O VRP (*Vehicle Routing Problem*) é um problema que consiste em definir rotas para um conjunto de veículos estacionados em um depósito central que irão servir um conjunto de clientes, minimizando os custos de transporte.
- Este problema corresponde a problemas de otimização no intuito de cobrir os nós de um grafo, contendo um nó representando o depósito, a mínimo custo.

Problema Roteamento de Veículos

- Os problemas reais são caracterizados por diversas restrições que limitam a tipologia dos ciclos e tornam o VRP um dos mais difíceis entre os problemas de otimização combinatorial.
- O VRP é tido com NP-hard e os algoritmos exatos propostos na literatura são capazes de resolver apenas problemas de menores dimensões e aparentemente não levam em consideração a complexidade de problemas reais.

Problema Roteamento de Veículos



https://neo.lcc.uma.es/vrp/vehicle-routing-problem/

Problema Roteamento de Veículos

- Algoritmo construtivo de Clarke-Wright para o VRP: Economias/*Savings*

Passo 1: rotule os clientes como nós 1,…,n e denomine o armazém de nó 0.
Passo 2: determine os custos c_{ij}, i = 0,1,…,n; j = 0,1,…,n para viajar entre todos os pares de cidades e o armazém
Passo 3: selecione o armazém como o nó central
Passo 4: Calcule as economias $s_{ij} = c_{i0} + c_{0j} - c_{ij}$, para i = 0,1,…,n; j = 0,1,…,n; i≠j.
Passo 5: Crie n rotas de veículos 0–i–0 para i = 0,1,…,n
Passo 6: Ordene as economias, s_{ij} , da maior à menor.
Passo 7: Tome a atual aresta (i,j) do topo da lista de economias
- **Passo 7.1:** **chamar função** para verificar se os nós i e j já estão em alguma outra rota e por isso possuem restrições para compartilharem da mesma rota.
- **Passo 7.2:** **chamar função** para verificar se as restrições estão sendo atendidas.
- **Passo 7.3:** Se passou em todas as verificações, os dois nós passam a fazer parte da nova rota.

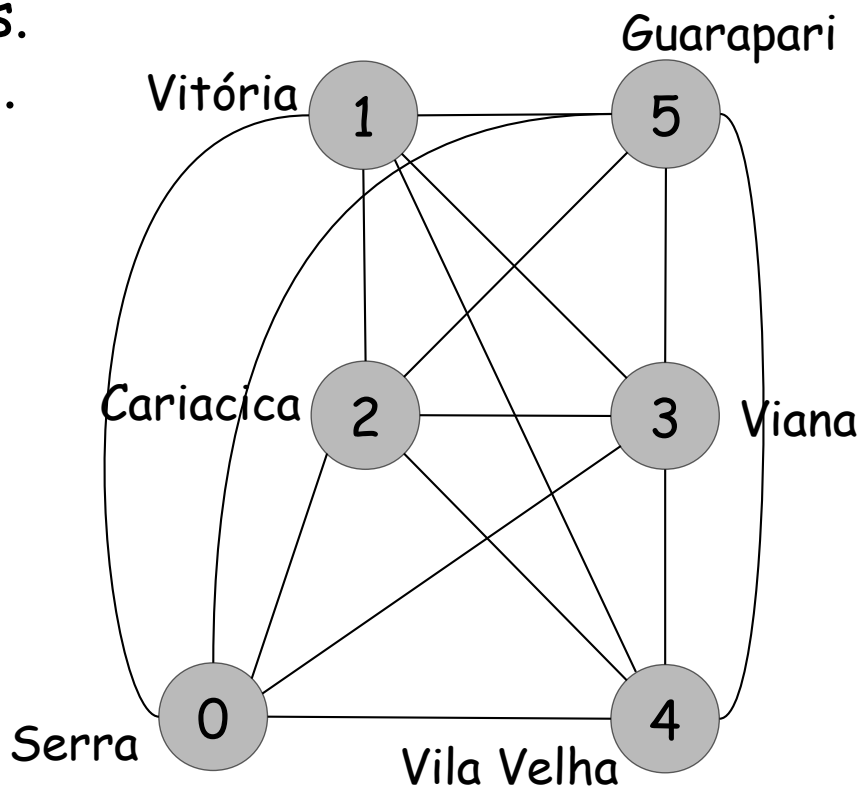
As rotas com a seqüência dos nós visitados é a saída do algoritmo.

Problema Roteamento de Veículos

- Algoritmo construtivo de Clarke-Wright para o VRP: Economias/*Savings*

Suponha o seguinte mapa de distribuição da empresa Wine, cujo centro de distribuição fica localizado na Serra e os clientes distribuídos pelos municípios da Grande Vitória e adjacências.
A capacidade do veículo é de 100 unidades.
A demanda do cliente é dada na tabela:

Cliente	Demanda (unidades)
Vitória (1)	37
Cariacica (2)	35
Viana (3)	30
Vila Velha (4)	25
Guarapari (5)	32

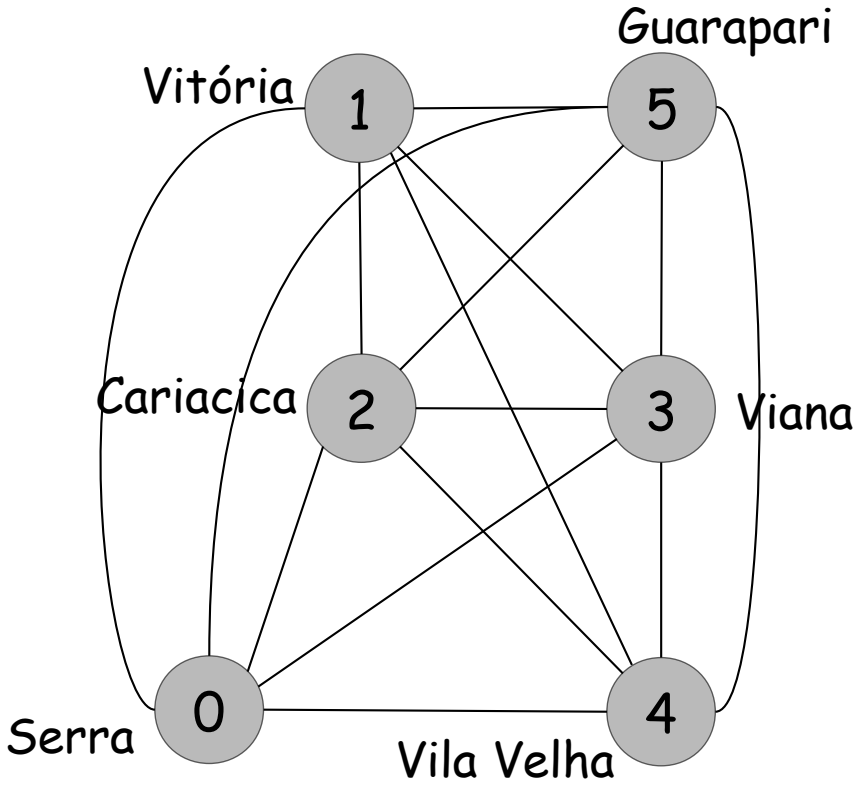


Problema Roteamento de Veículos

- Algoritmo construtivo de Clarke-Wright para o VRP: Economias/*Savings*

Passo 1: rotule os clientes como nós 1,…,n e denomine o **armazém de nó 0**.

Cliente	Demanda (unidades)
Vitória (1)	37
Cariacica (2)	35
Viana (3)	30
Vila Velha (4)	25
Guarapari (5)	32

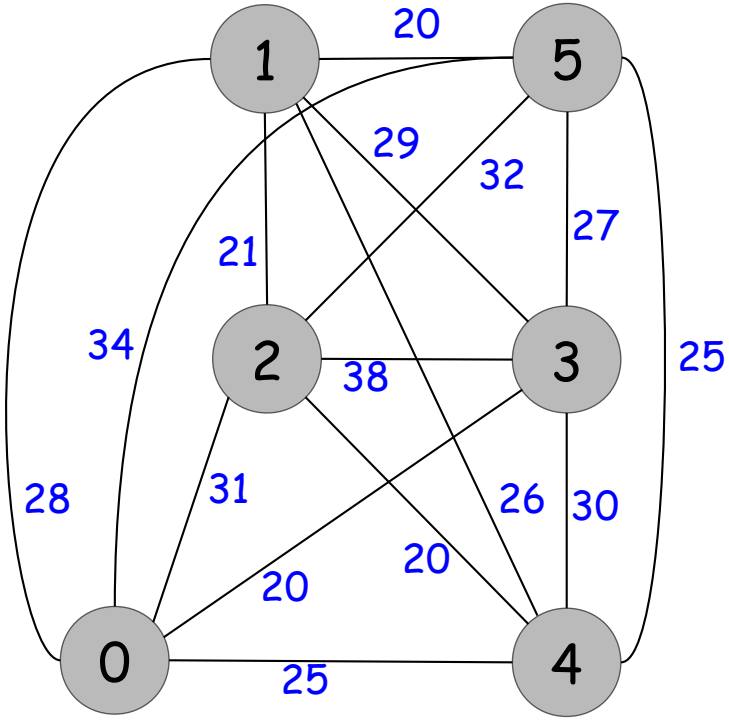


Problema Roteamento de Veículos

- Algoritmo construtivo de Clarke-Wright para o VRP: Economias/*Savings*

Passo 1: rotule os clientes como nós 1,…,n e denomine o armazém de nó 0.
Passo 2: determine os custos c_{ij}, i = 0,1,…,n; j = 0,1,…,n para viajar entre todos os pares de cidades e o armazém

C_{ij}	0	1	2	3	4	5
0	-	28	31	20	25	34
1		-	21	29	26	20
2			-	38	20	32
3				-	30	27
4					-	25
5						-

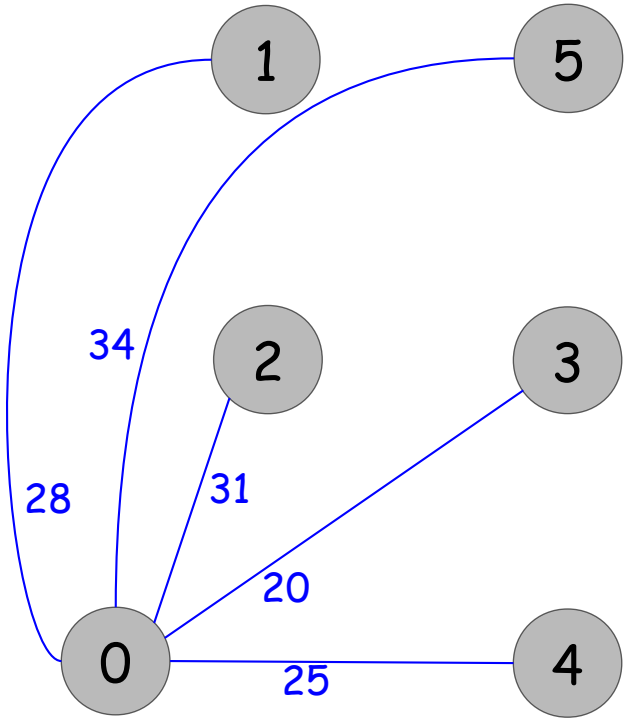


Problema Roteamento de Veículos

- Algoritmo construtivo de Clarke-Wright para o VRP: Economias/*Savings*

Passo 1: rotule os clientes como nós 1,…,n e denomine o armazém de nó 0.
Passo 2: determine os custos c_{ij}, i = 0,1,…,n; j = 0,1,…,n para viajar entre todos os pares de cidades e o armazém

C_{ij}	0	1	2	3	4	5
0	-	28	31	20	25	34
1		-	21	29	26	20
2			-	38	20	32
3				-	30	27
4					-	25
5						-

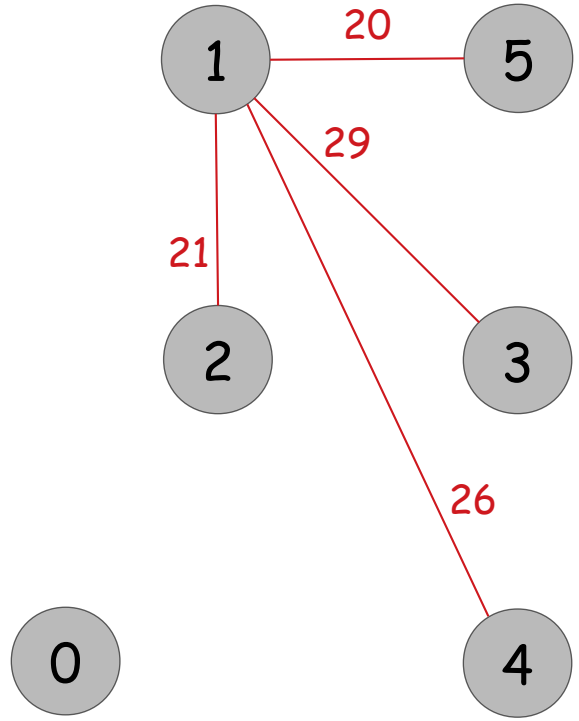


Problema Roteamento de Veículos

- Algoritmo construtivo de Clarke-Wright para o VRP: Economias/*Savings*

Passo 1: rotule os clientes como nós 1,…,n e denomine o armazém de nó 0.
Passo 2: determine os custos c_{ij}, i = 0,1,…,n; j = 0,1,…,n para viajar entre todos os pares de cidades e o armazém

C_{ij}	0	1	2	3	4	5
0	-	28	31	20	25	34
1		-	21	29	26	20
2			-	38	20	32
3				-	30	27
4					-	25
5						-

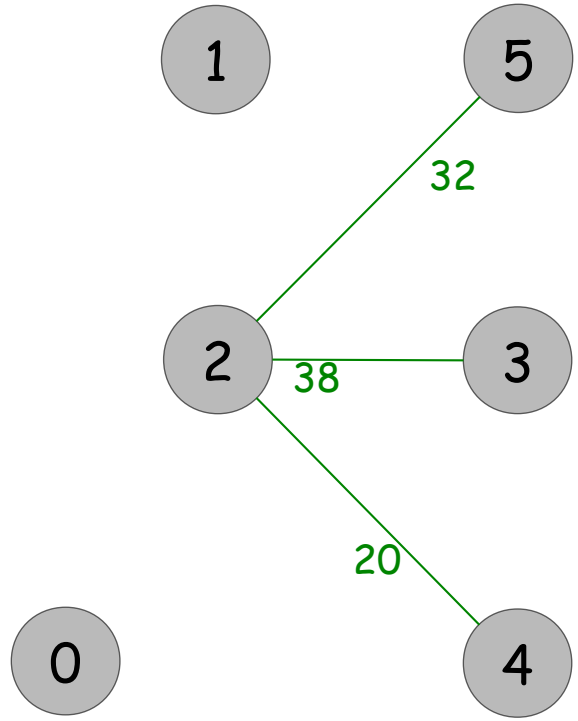


Problema Roteamento de Veículos

- Algoritmo construtivo de Clarke-Wright para o VRP: Economias/*Savings*

Passo 1: rotule os clientes como nós 1,…,n e denomine o armazém de nó 0.
Passo 2: determine os custos c_{ij}, i = 0,1,…,n; j = 0,1,…,n para viajar entre todos os pares de cidades e o armazém

C_{ij}	0	1	2	3	4	5
0	-	28	31	20	25	34
1		-	21	29	26	20
2			-	38	20	32
3				-	30	27
4					-	25
5						-

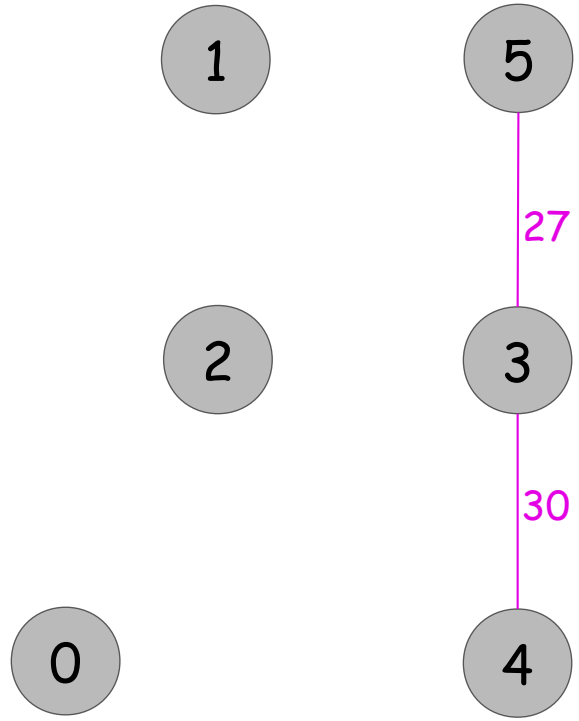


Problema Roteamento de Veículos

- Algoritmo construtivo de Clarke-Wright para o VRP: Economias/*Savings*

Passo 1: rotule os clientes como nós 1,…,n e denomine o armazém de nó 0.
Passo 2: determine os custos c_{ij}, i = 0,1,…,n; j = 0,1,…,n para viajar entre todos os pares de cidades e o armazém

C_{ij}	0	1	2	3	4	5
0	-	28	31	20	25	34
1		-	21	29	26	20
2			-	38	20	32
3				-	30	27
4					-	25
5						-

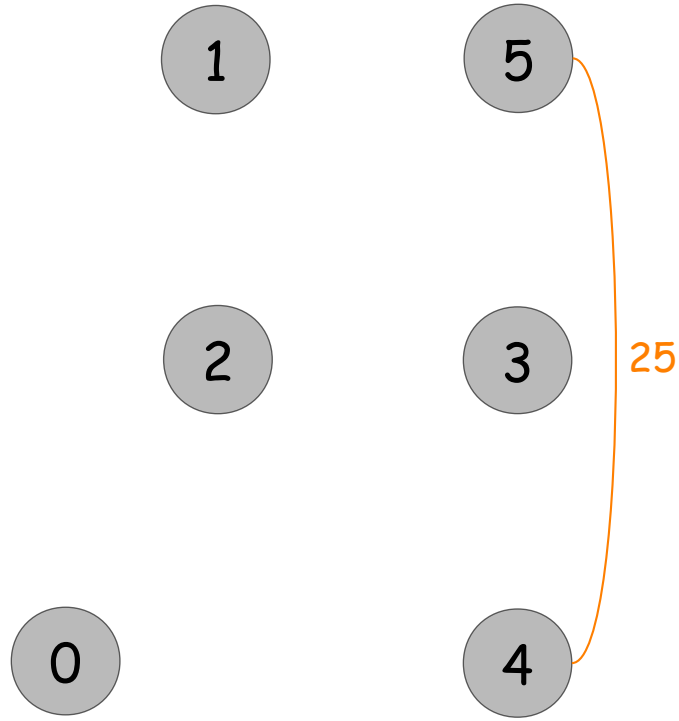


Problema Roteamento de Veículos

- Algoritmo construtivo de Clarke-Wright para o VRP: Economias/*Savings*

Passo 1: rotule os clientes como nós 1,…,n e denomine o armazém de nó 0.
Passo 2: determine os custos c_{ij}, i = 0,1,…,n; j = 0,1,…,n para viajar entre todos os pares de cidades e o armazém

C_{ij}	0	1	2	3	4	5
0	-	28	31	20	25	34
1		-	21	29	26	20
2			-	38	20	32
3				-	30	27
4					-	25
5						-

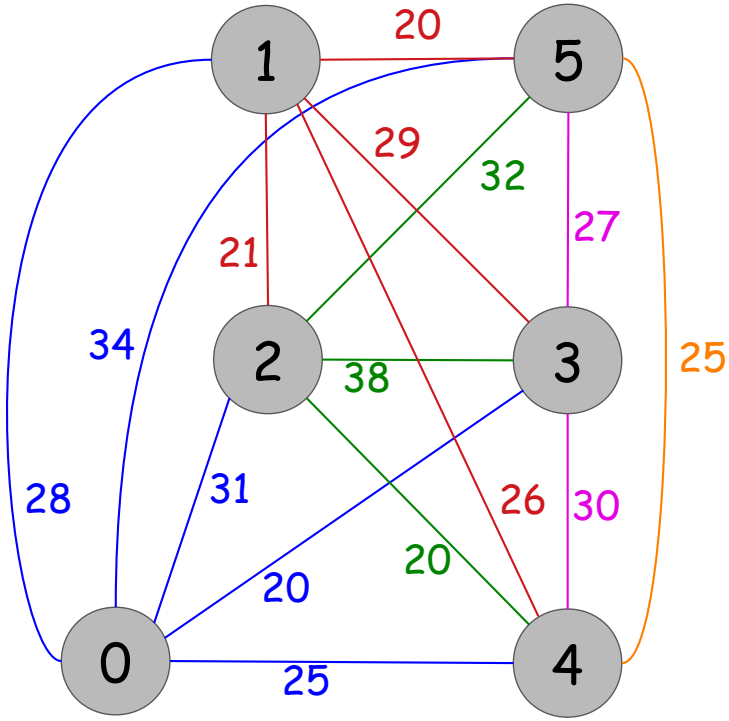


Problema Roteamento de Veículos

- Algoritmo construtivo de Clarke-Wright para o VRP: Economias/*Savings*

Passo 1: rotule os clientes como nós 1,…,n e denomine o armazém de nó 0.
Passo 2: determine os custos c_{ij}, i = 0,1,…,n; j = 0,1,…,n para viajar entre todos os pares de cidades e o armazém

C_{ij}	0	1	2	3	4	5
0	-	28	31	20	25	34
1		-	21	29	26	20
2			-	38	20	32
3				-	30	27
4					-	25
5						-

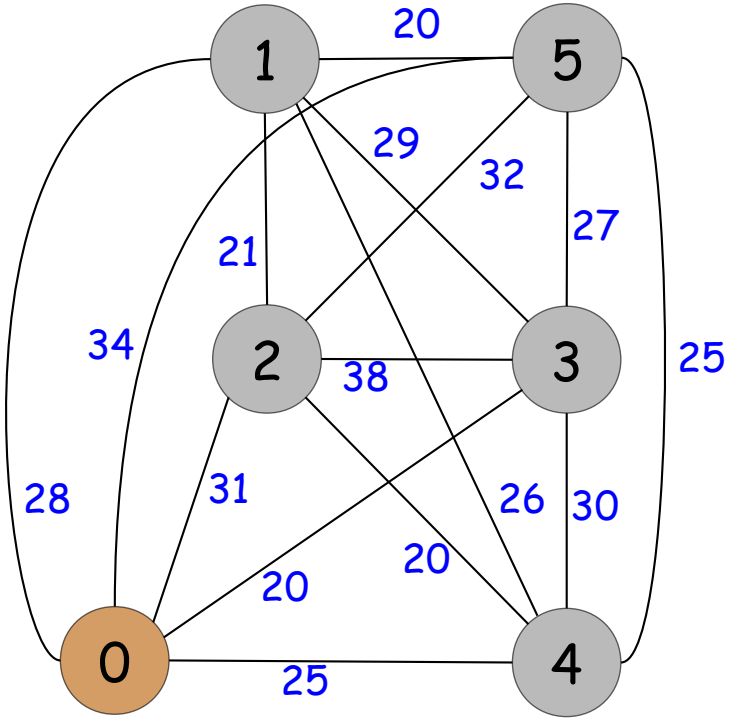


Problema Roteamento de Veículos

- Algoritmo construtivo de Clarke-Wright para o VRP: Economias/*Savings*

Passo 1: rotule os clientes como nós 1,…,n e denomine o armazém de nó 0.
Passo 2: determine os custos c_{ij}, i = 0,1,…,n; j = 0,1,…,n para viajar entre todos os pares de cidades e o armazém
Passo 3: selecione o armazém como o nó central

C_{ij}	0	1	2	3	4	5
0	-	28	31	20	25	34
1		-	21	29	26	20
2			-	38	20	32
3				-	30	27
4					-	25
5						-



Problema Roteamento de Veículos

- Algoritmo construtivo de Clarke-Wright para o VRP: Economias/*Savings*

Passo 4: Calcule as economias $s_{ij} = c_{i0} + c_{0j} - c_{ij}$, para i = 0,1,…,n; j = 0,1,…,n; i≠j.

$$s_{23} = c_{20} + c_{03} - c_{23}$$
$$13 = 31 + 20 - 38$$

C_{ij}	0	1	2	3	4	5
0	-	28	31	20	25	34
1		-	21	29	26	20
2			-	38	20	32
3				-	30	27
4					-	25
5						-

S_{ij}	1	2	3	4	5
1	-	38	19	27	42
2		-	13	36	33
3			-	15	27
4				-	34
5					-

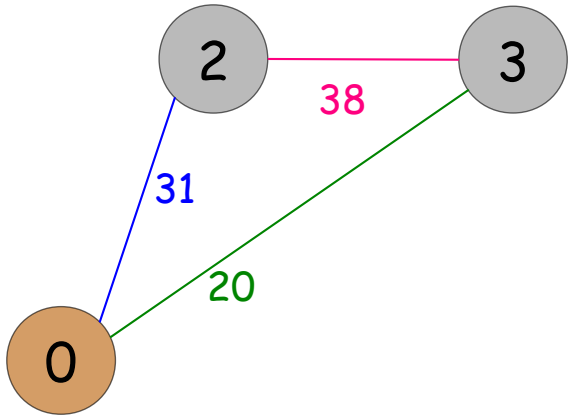
Problema Roteamento de Veículos

S_{ij}	1	2	3	4	5
1	-	38	19	27	42
2		-	13	36	33
3			-	15	27
4				-	34
5					-

$$s_{23} = c_{20} + c_{03} - c_{23}$$

$$13 = 31 + 20 - 38$$

C_{ij}	0	1	2	3	4	5
0	-	28	31	20	25	34
1		-	21	29	26	20
2			-	38	20	32
3				-	30	27
4					-	25
5						-



Problema Roteamento de Veículos

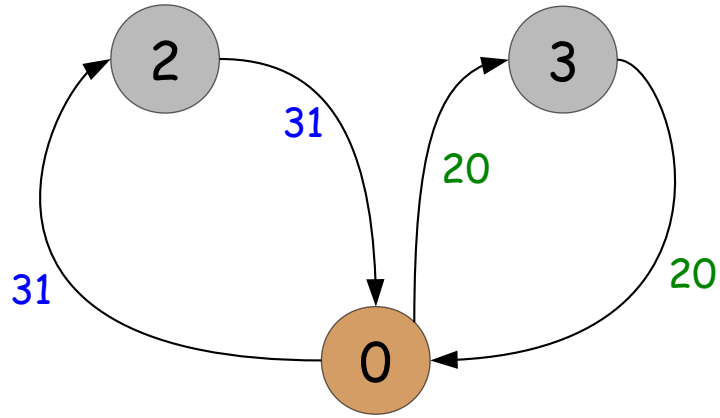
S_{ij}	1	2	3	4	5
1	-	38	19	27	42
2		-	13	36	33
3			-	15	27
4				-	34
5					-

$$D_{020} = 31 + 31 = 62$$

$$D_{030} = 20 + 20 = 40$$

$$D_{020} + D_{030} = 102$$

C_{ij}	0	1	2	3	4	5
0	-	28	31	20	25	34
1		-	21	29	26	20
2			-	38	20	32
3				-	30	27
4					-	25
5						-



Problema Roteamento de Veículos

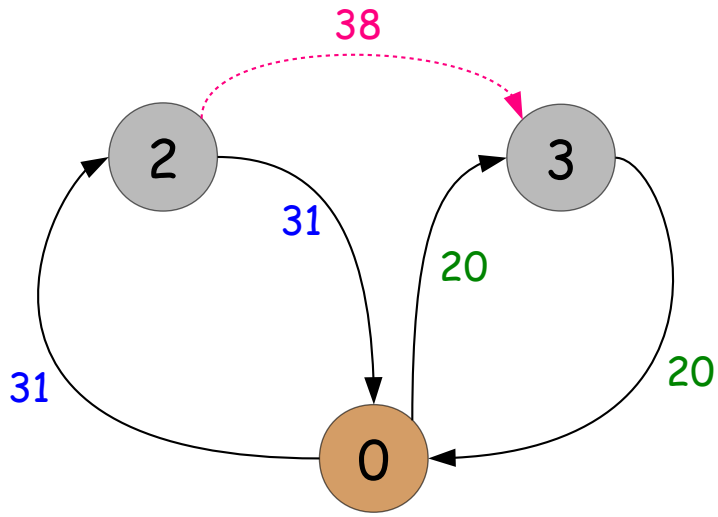
S_{ij}	1	2	3	4	5
1	-	38	19	27	42
2		-	13	36	33
3			-	15	27
4				-	34
5					-

$$D_{020} = 31 + 31 = 62$$

$$D_{030} = 20 + 20 = 40$$

$$D_{020} + D_{030} = 102$$

C_{ij}	0	1	2	3	4	5
0	-	28	31	20	25	34
1		-	21	29	26	20
2			-	38	20	32
3				-	30	27
4					-	25
5						-



Problema Roteamento de Veículos

S_{ij}	1	2	3	4	5
1	-	38	19	27	42
2		-	13	36	33
3			-	15	27
4				-	34
5					-

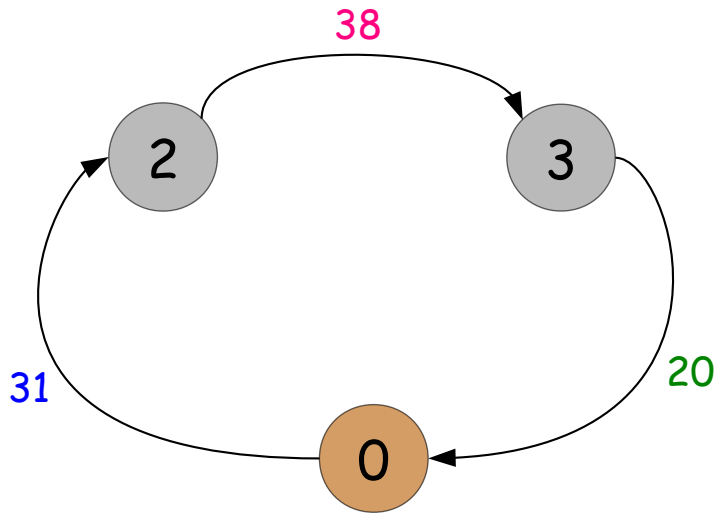
$D_{02} = 31$

$D_{23} = 38$

$D_{30} = 20$

$D_{0230} = 89$

C_{ij}	0	1	2	3	4	5
0	-	28	31	20	25	34
1		-	21	29	26	20
2			-	38	20	32
3				-	30	27
4					-	25
5						-



Problema Roteamento de Veículos

$D_{020} = 31 + 31 = 62$

$D_{030} = 20 + 20 = 40$

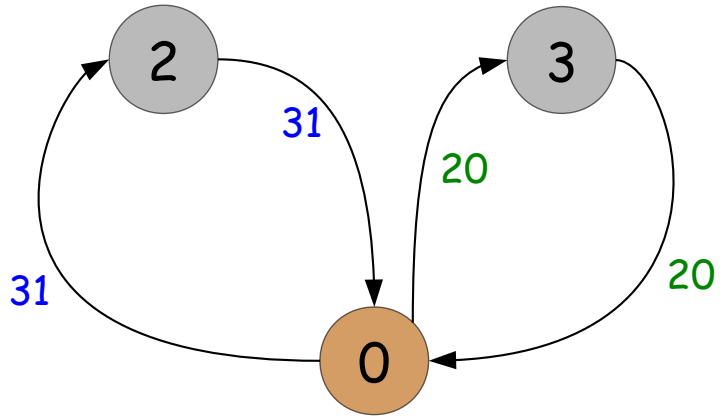
$D_{020} + D_{030} = 102$

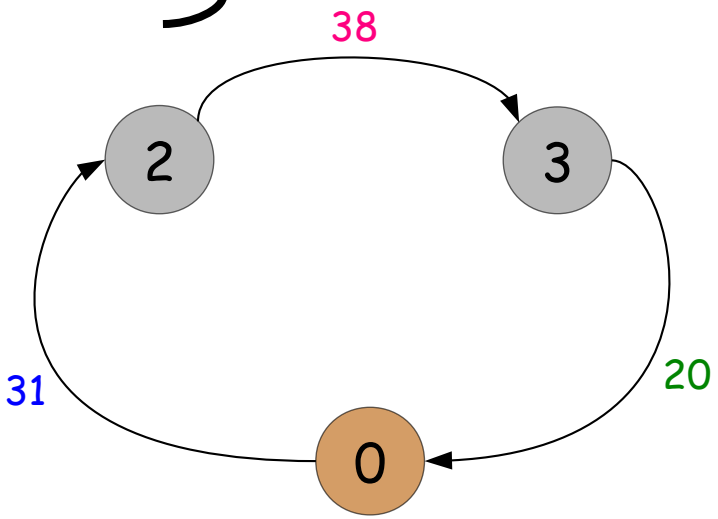
$D_{02} = 31$

$D_{23} = 38$

$D_{30} = 20$

$D_{0230} = 89$





$s_{23} = c_{20} + c_{03} - c_{23}$

$13 = 31 + 20 - 38$

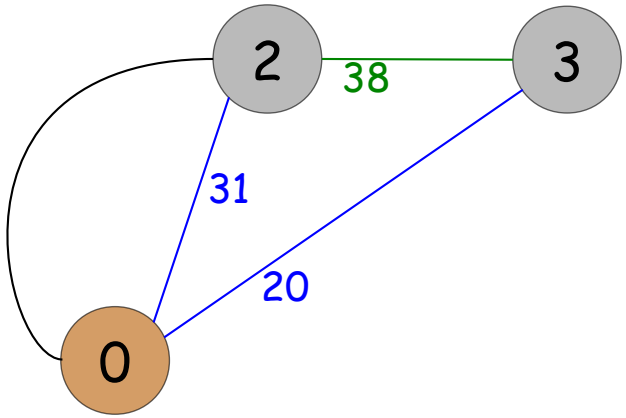
Problema Roteamento de Veículos

S_{ij}	1	2	3	4	5
1	-	38	19	27	42
2		-	13	36	33
3			-	15	27
4				-	34
5					-

$$s_{23} = c_{20} + c_{03} - c_{23}$$

$$13 = 31 + 20 - 38$$

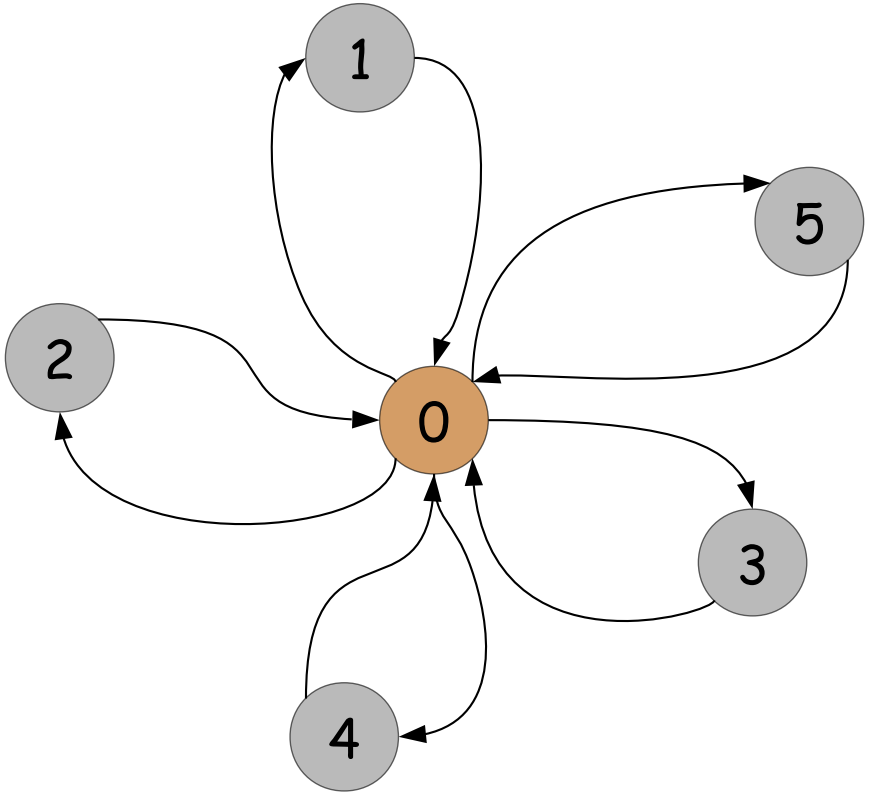
C_{ij}	0	1	2	3	4	5
0	-	28	31	20	25	34
1		-	21	29	26	20
2			-	38	20	32
3				-	30	27
4					-	25
5						-



Problema Roteamento de Veículos

- Algoritmo construtivo de Clarke-Wright para o VRP: Economias/*Savings*

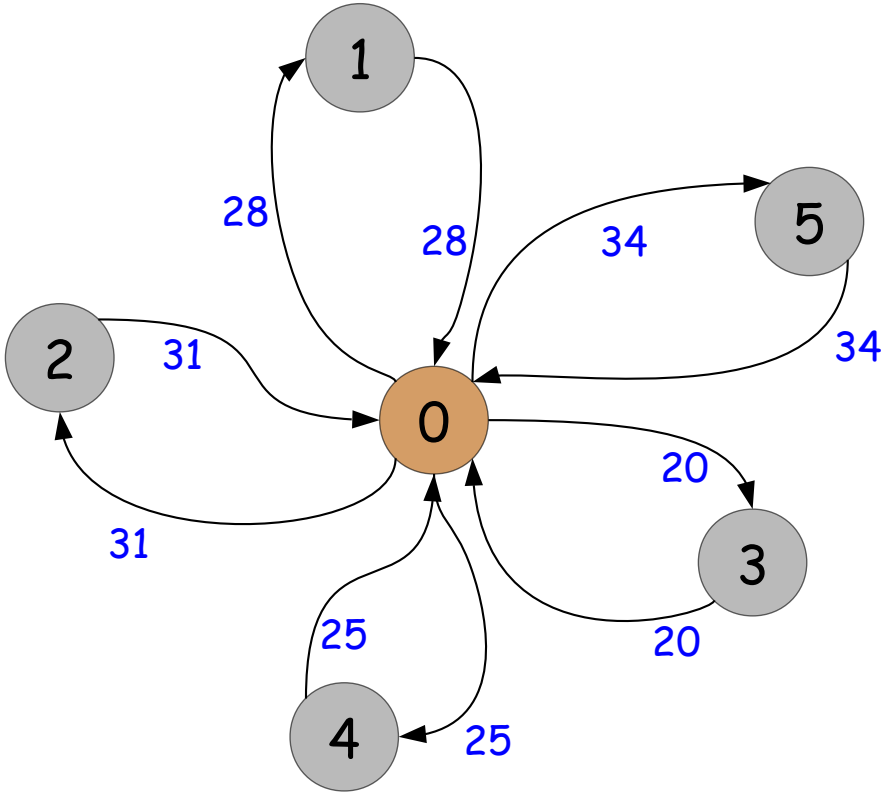
Passo 5: Crie n rotas de veículos 0–i–0 para i = 0,1,…,n



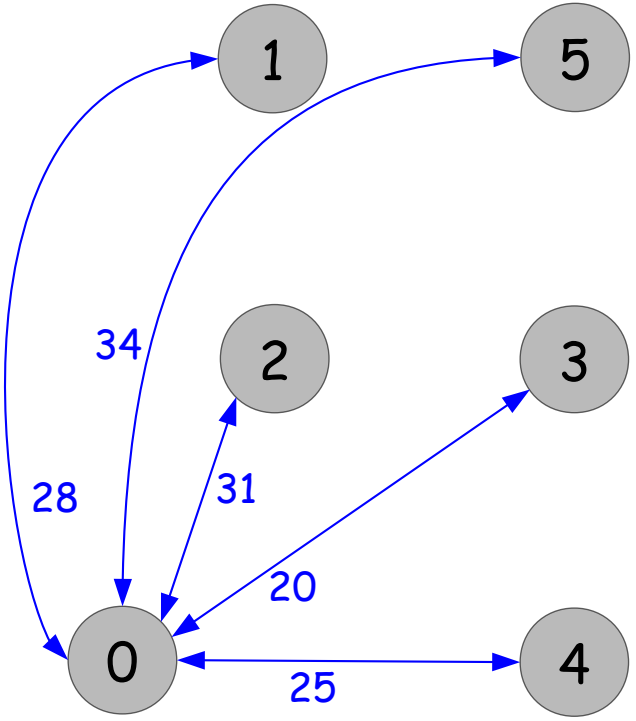
Problema Roteamento de Veículos

- Algoritmo construtivo de Clarke-Wright para o VRP: Economias/*Savings*

Passo 5: Crie n rotas de veículos 0–i–0 para i = 0,1,…,n



São necessários 5 veículos ao custo total de **276**



Problema Roteamento de Veículos

- Algoritmo construtivo de Clarke-Wright para o VRP: Economias/*Savings*

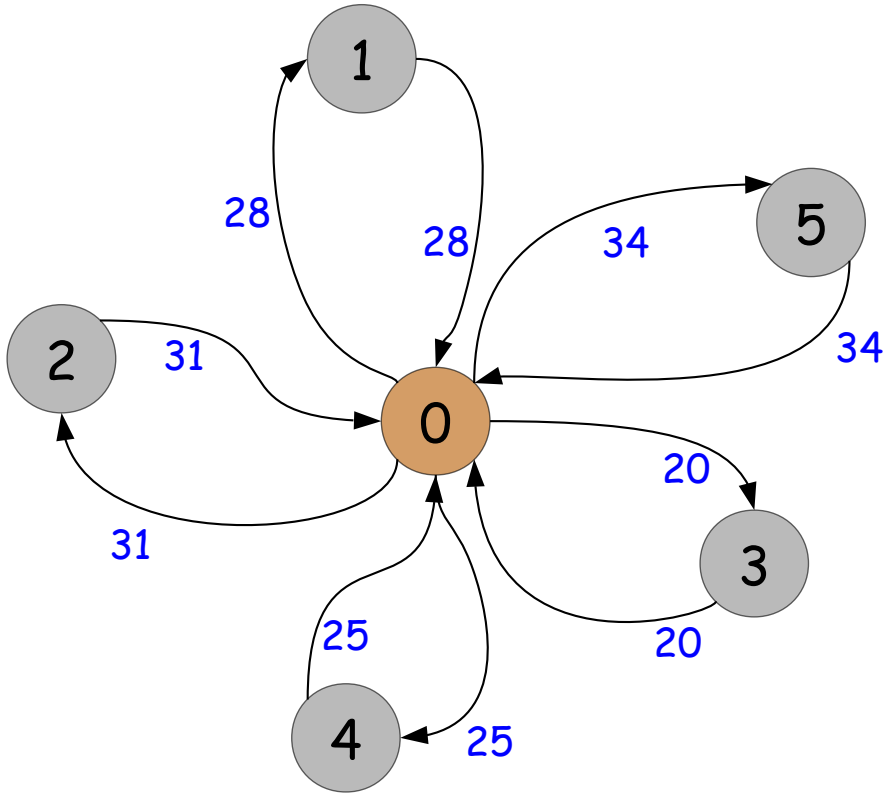
Passo 6: Ordene as economias, s_{ij}, da maior à menor (decrescente).

S_{ij}	1	2	3	4	5
1	-	38	19	27	42
2		-	13	36	33
3			-	15	27
4				-	34
5					-

	i-j	S_{ij}
S_{15}	1-5	42
S_{12}	1-2	38
S_{24}	2-4	36
S_{45}	4-5	34
S_{25}	2-5	33
S_{14}	1-4	27
S_{35}	3-5	27
S_{13}	1-3	19
S_{34}	3-4	15
S_{23}	2-3	13

Problema Roteamento de Veículos

Passo 7: Tome a atual aresta (i,j) do topo da lista de economias
Passo 7.1: **função** de verificar se nós i e j já estão em alguma outra rota
Passo 7.2: **função** para verificar se as restrições estão sendo atendidas.
Passo 7.3: Se passou em 7.1 e 7.2, i e j passam a fazer parte da nova rota.



S_{15}	42
S_{12}	38
S_{24}	36
S_{45}	34
S_{25}	33
S_{14}	27
S_{35}	27
S_{13}	19
S_{34}	15
S_{23}	13

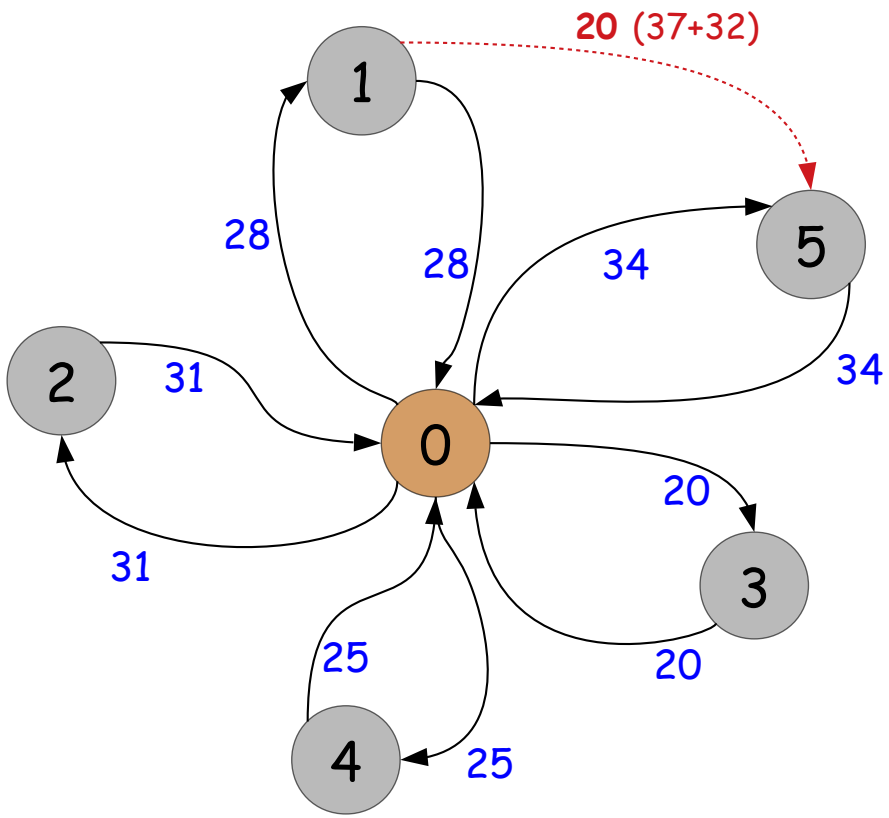
Problema Roteamento de Veículos

Passo 7: **Tome a atual aresta (i,j) do topo da lista de economias**

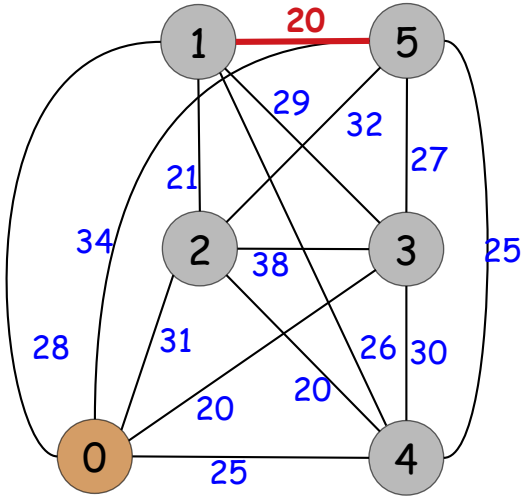
Passo 7.1: **função** de verificar se nós i e j já estão em alguma outra rota

Passo 7.2: **função** para verificar se as restrições estão sendo atendidas.

Passo 7.3: Se passou em 7.1 e 7.2, i e j passam a fazer parte da nova rota.



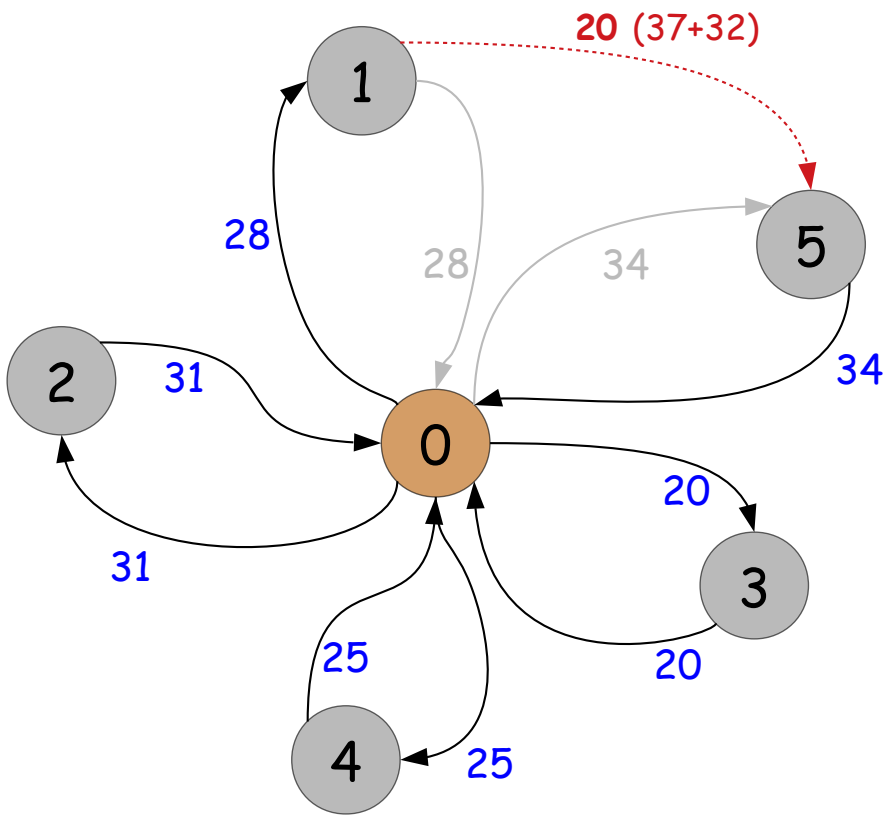
S_{15}	42
S_{12}	38
S_{24}	36
S_{45}	34
S_{25}	33
S_{14}	27
S_{35}	27
S_{13}	19
S_{34}	15
S_{23}	13



Demandas	
1	37
2	35
3	30
4	25
5	32

Problema Roteamento de Veículos

Passo 7: Tome a atual aresta (i,j) do topo da lista de economias
Passo 7.1: **função de verificar se nós i e j já estão em alguma outra rota**
Passo 7.2: **função** para verificar se as restrições estão sendo atendidas.
Passo 7.3: Se passou em 7.1 e 7.2, i e j passam a fazer parte da nova rota.

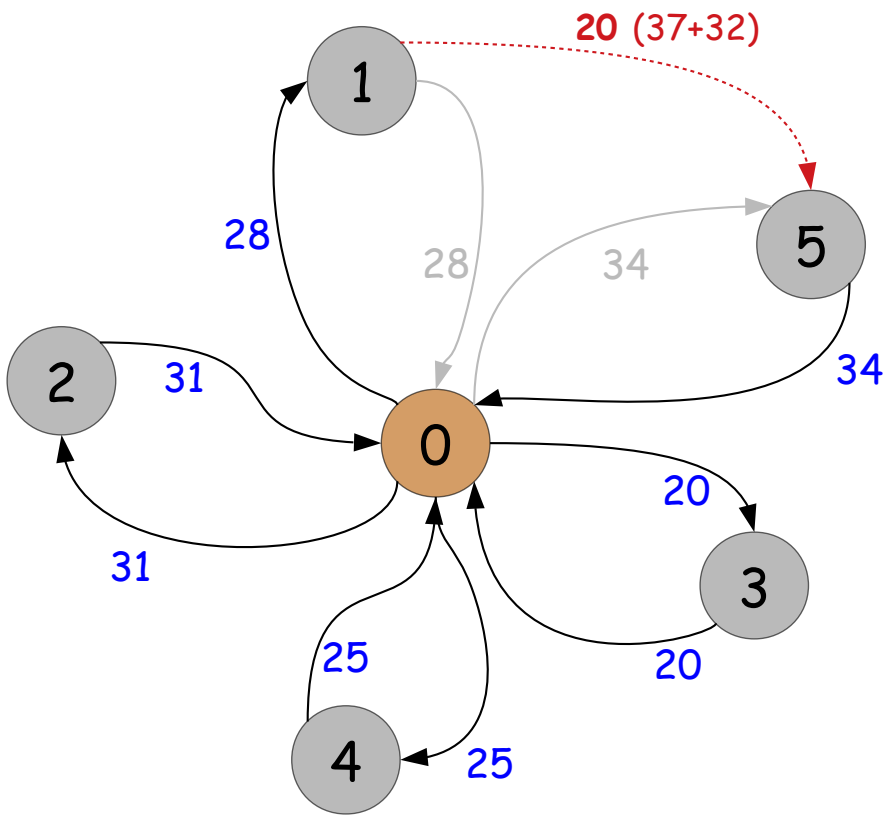


S_{15}	42
S_{12}	38
S_{24}	36
S_{45}	34
S_{25}	33
S_{14}	27
S_{35}	27
S_{13}	19
S_{34}	15
S_{23}	13

OK! Rotas podem ser unidadas

Problema Roteamento de Veículos

Passo 7: Tome a atual aresta (i,j) do topo da lista de economias
Passo 7.1: função de verificar se nós i e j já estão em alguma outra rota
Passo 7.2: **função para verificar se as restrições estão sendo atendidas.**
Passo 7.3: Se passou em 7.1 e 7.2, i e j passam a fazer parte da nova rota.

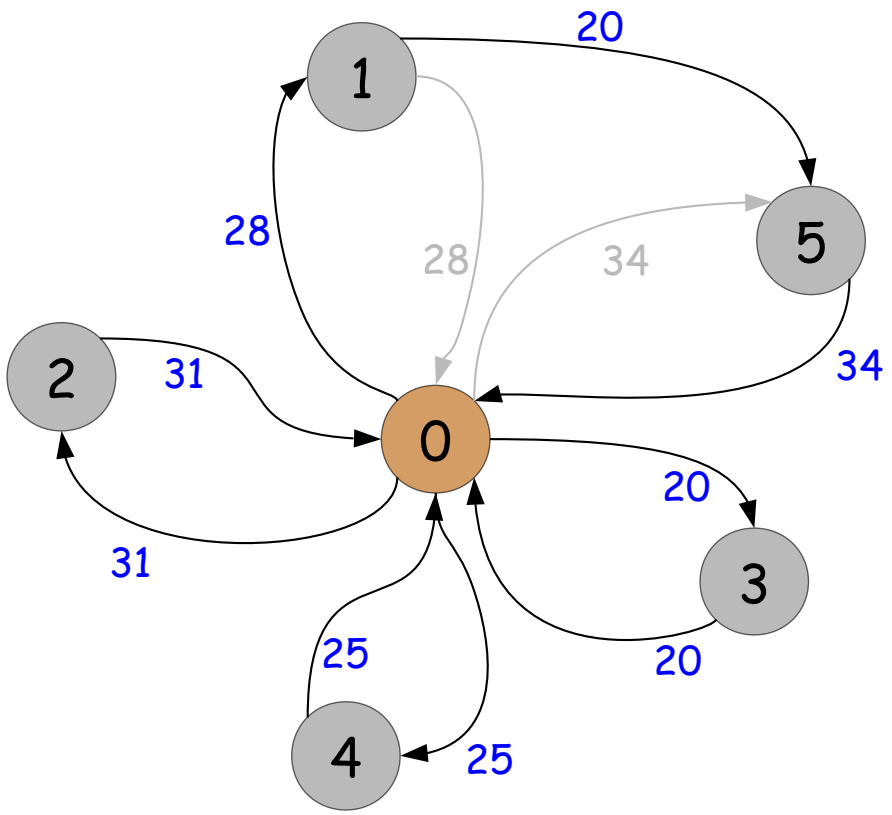


S_{15}	42
S_{12}	38
S_{24}	36
S_{45}	34
S_{25}	33
S_{14}	27
S_{35}	27
S_{13}	19
S_{34}	15
S_{23}	13

OK!
Única restrição é limite do veículo = 100
Demanda = 37+32 < 100
Demanda = 69 < 100

Problema Roteamento de Veículos

Passo 7: Tome a atual aresta (i,j) do topo da lista de economias
Passo 7.1: função de verificar se nós i e j já estão em alguma outra rota
Passo 7.2: função para verificar se as restrições estão sendo atendidas.
Passo 7.3: **Se passou em 7.1 e 7.2, i e j passam a fazer parte da nova rota.**

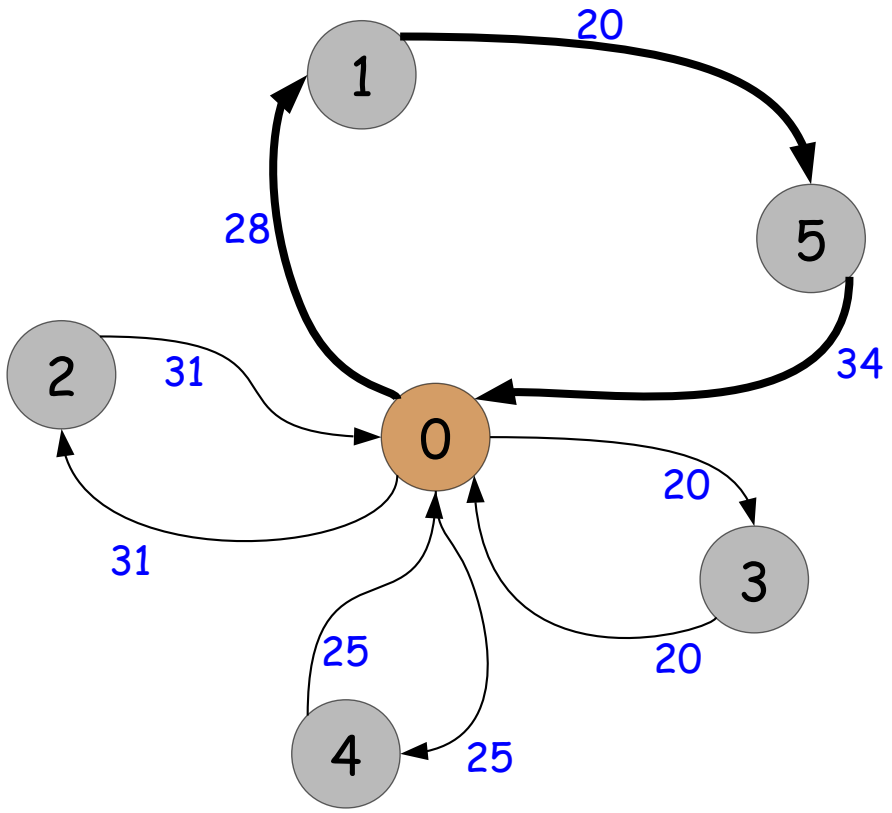


~~S_{15}~~	~~42~~
S_{12}	38
S_{24}	36
S_{45}	34
S_{25}	33
S_{14}	27
S_{35}	27
S_{13}	19
S_{34}	15
S_{23}	13

São necessários agora 4 veículos ao custo total de **276-42 = 234**

Problema Roteamento de Veículos

Passo 7: Tome a atual aresta (i,j) do topo da lista de economias
Passo 7.1: função de verificar se nós i e j já estão em alguma outra rota
Passo 7.2: função para verificar se as restrições estão sendo atendidas.
Passo 7.3: **Se passou em 7.1 e 7.2, i e j passam a fazer parte da nova rota.**



~~S_{15}~~	~~42~~
S_{12}	38
S_{24}	36
S_{45}	34
S_{25}	33
S_{14}	27
S_{35}	27
S_{13}	19
S_{34}	15
S_{23}	13

São necessários agora 4 veículos ao custo total de **276-42 = 234**

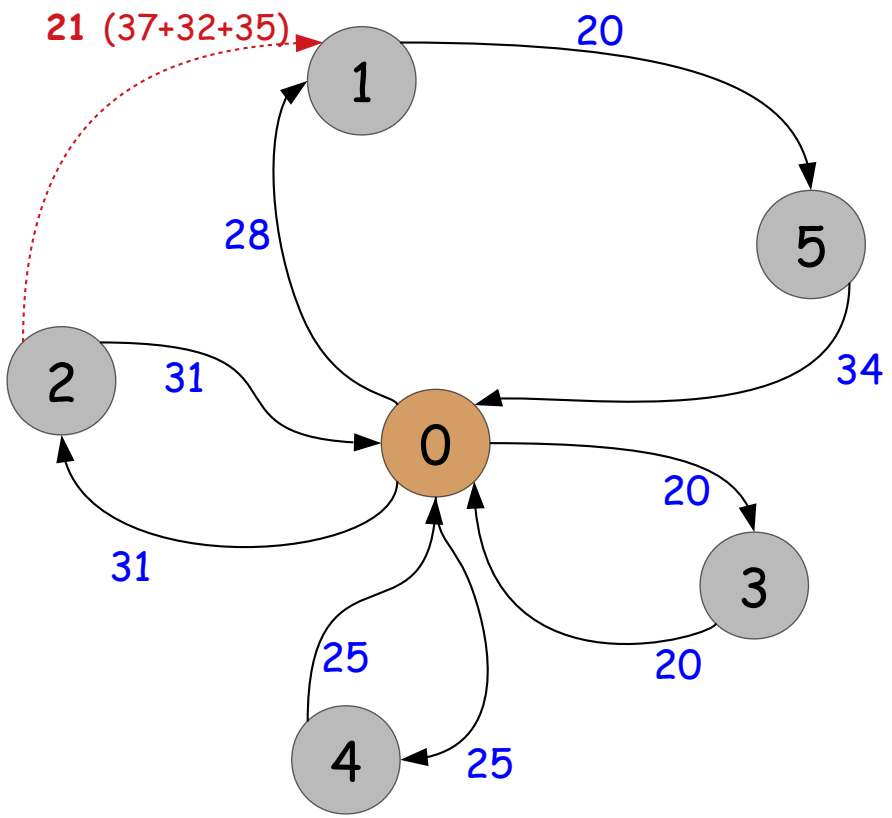
Problema Roteamento de Veículos

Passo 7: **Tome a atual aresta (i,j) do topo da lista de economias**

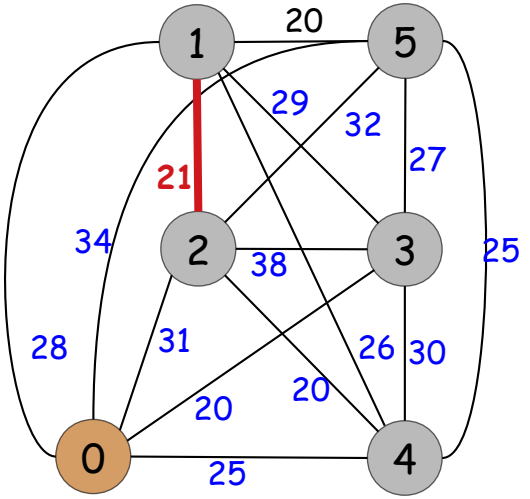
Passo 7.1: função de verificar se nós i e j já estão em alguma outra rota

Passo 7.2: função para verificar se as restrições estão sendo atendidas.

Passo 7.3: Se passou em 7.1 e 7.2, i e j passam a fazer parte da nova rota.



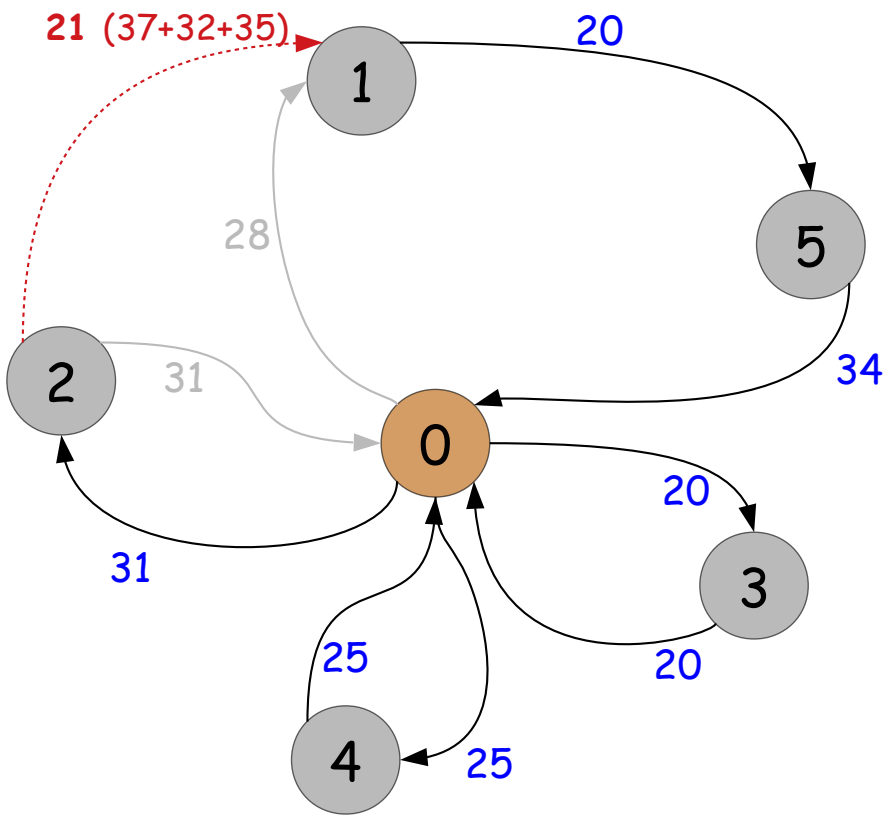
~~S_{15}~~	~~42~~
S_{12}	38
S_{24}	36
S_{45}	34
S_{25}	33
S_{14}	27
S_{35}	27
S_{13}	19
S_{34}	15
S_{23}	13



Demandas	
1	37
2	35
3	30
4	25
5	32

Problema Roteamento de Veículos

Passo 7: Tome a atual aresta (i,j) do topo da lista de economias
Passo 7.1: **função de verificar se nós i e j já estão em alguma outra rota**
Passo 7.2: **função** para verificar se as restrições estão sendo atendidas.
Passo 7.3: Se passou em 7.1 e 7.2, i e j passam a fazer parte da nova rota.

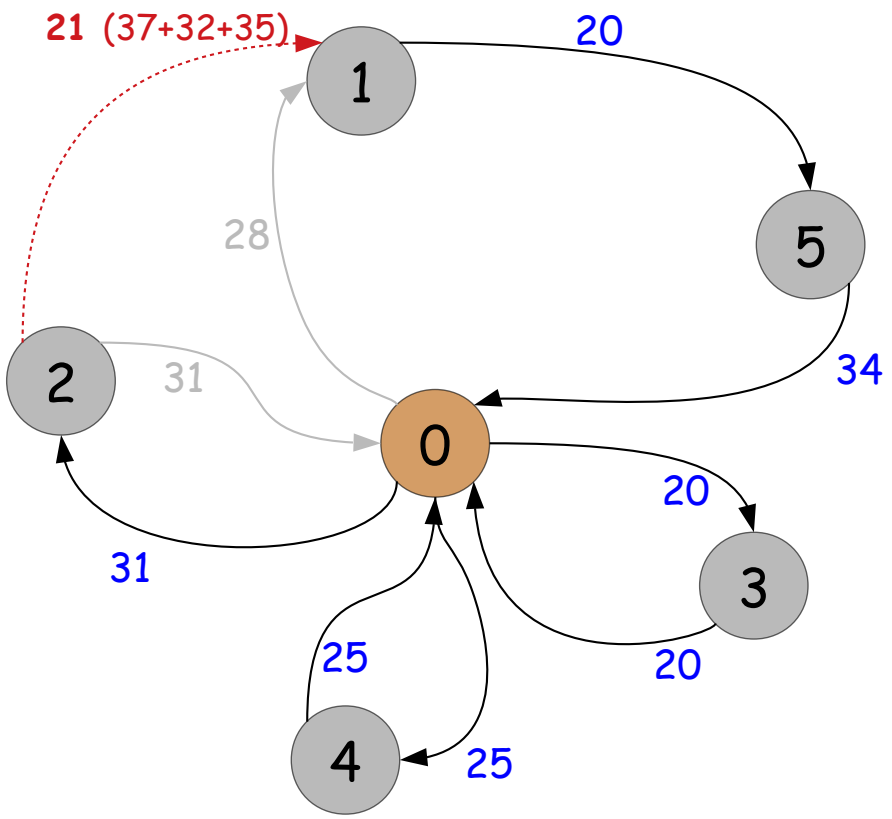


~~S_{15}~~	~~42~~
S_{12}	**38**
S_{24}	36
S_{45}	34
S_{25}	33
S_{14}	27
S_{35}	27
S_{13}	19
S_{34}	15
S_{23}	13

OK! Rotas podem ser unidadas

Problema Roteamento de Veículos

Passo 7: Tome a atual aresta (i,j) do topo da lista de economias
Passo 7.1: função de verificar se nós i e j já estão em alguma outra rota
Passo 7.2: **função para verificar se as restrições estão sendo atendidas.**
Passo 7.3: Se passou em 7.1 e 7.2, i e j passam a fazer parte da nova rota.

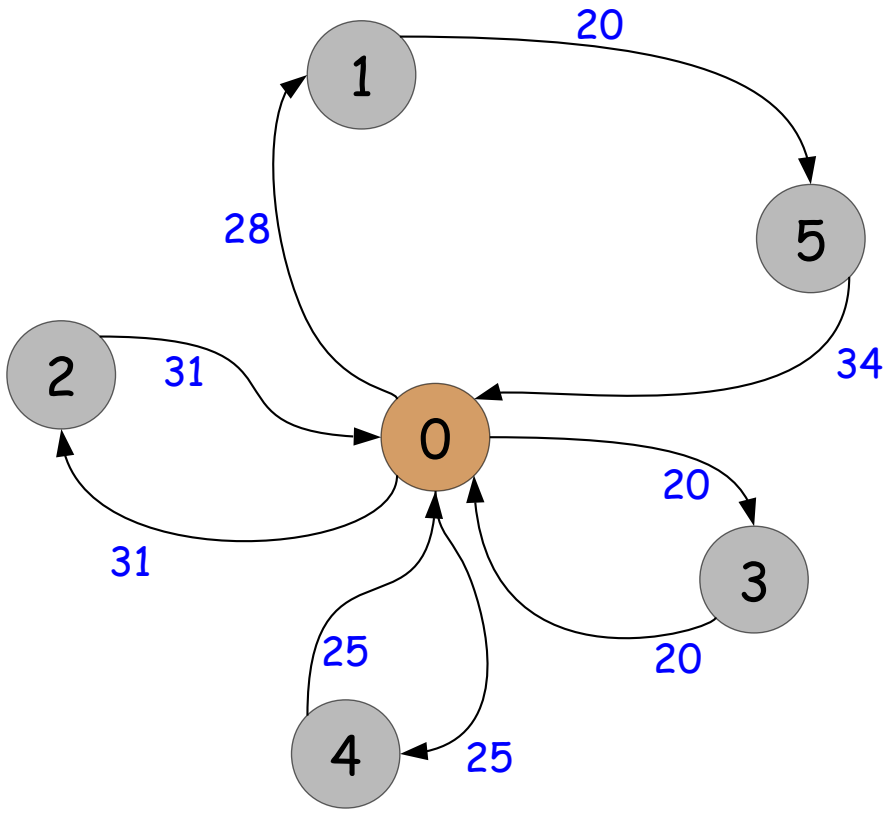


~~S_{15}~~	~~42~~
S_{12}	**38**
S_{24}	36
S_{45}	34
S_{25}	33
S_{14}	27
S_{35}	27
S_{13}	19
S_{34}	15
S_{23}	13

FALHOU!
Única restrição é limite do veículo = 100
Demanda = 37+32+35 > 100
Demanda = 104 > 100

Problema Roteamento de Veículos

Passo 7: Tome a atual aresta (i,j) do topo da lista de economias
Passo 7.1: função de verificar se nós i e j já estão em alguma outra rota
Passo 7.2: função para verificar se as restrições estão sendo atendidas.
Passo 7.3: **Se passou em 7.1 e 7.2, i e j passam a fazer parte da nova rota.**



~~S_{15}~~	~~42~~
~~S_{12}~~	~~38~~
S_{24}	36
S_{45}	34
S_{25}	33
S_{14}	27
S_{35}	27
S_{13}	19
S_{34}	15
S_{23}	13

FALHOU!
Descarta aresta.

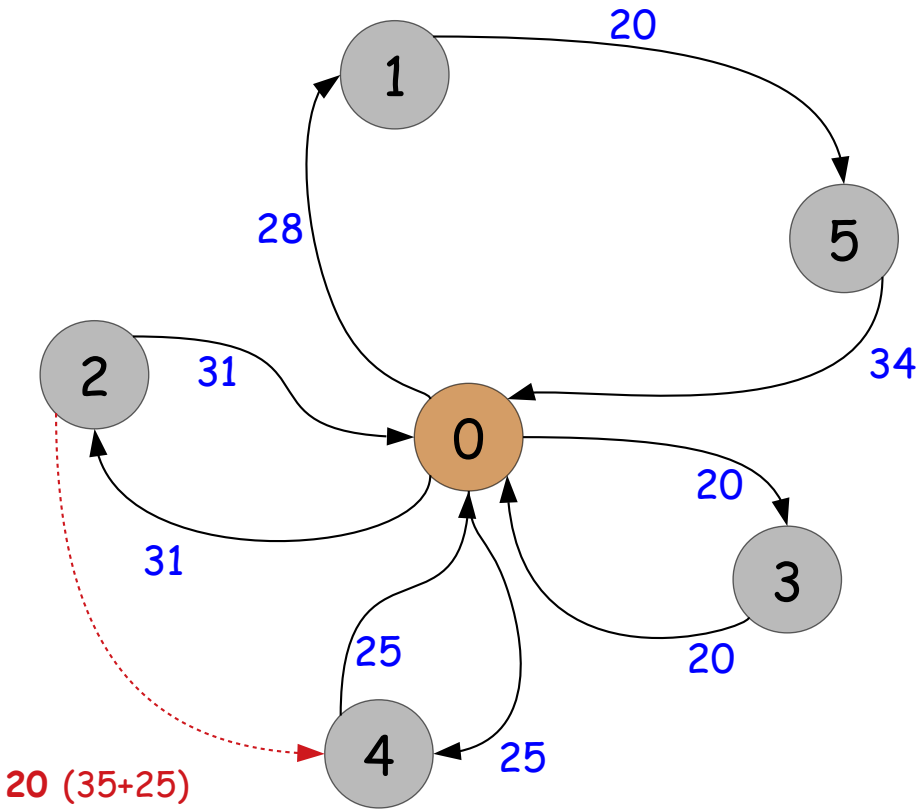
Problema Roteamento de Veículos

Passo 7: **Tome a atual aresta (i,j) do topo da lista de economias**

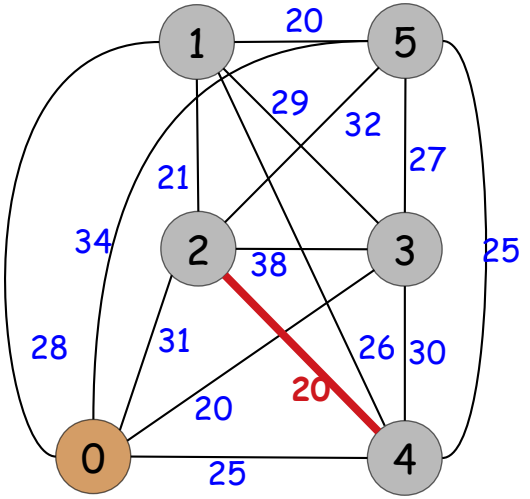
Passo 7.1: função de verificar se nós i e j já estão em alguma outra rota

Passo 7.2: função para verificar se as restrições estão sendo atendidas.

Passo 7.3: Se passou em 7.1 e 7.2, i e j passam a fazer parte da nova rota.



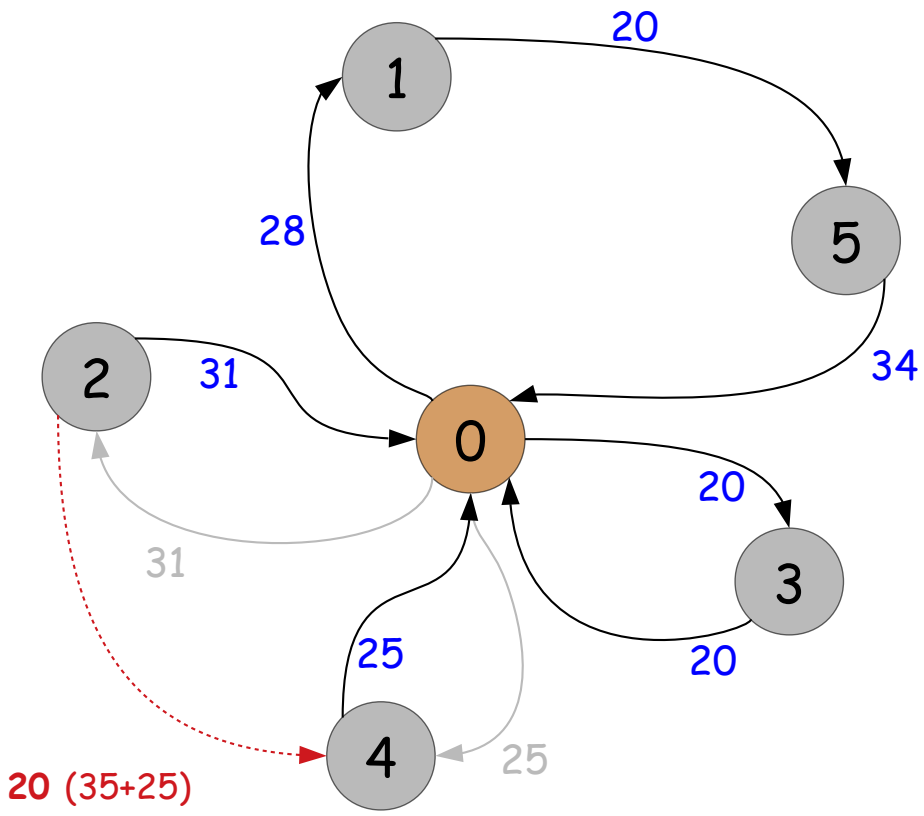
~~S_{15}~~	~~42~~
~~S_{12}~~	~~38~~
S_{24}	**36**
S_{45}	34
S_{25}	33
S_{14}	27
S_{35}	27
S_{13}	19
S_{34}	15
S_{23}	13



Demandas	
1	37
2	**35**
3	30
4	**25**
5	32

Problema Roteamento de Veículos

Passo 7: Tome a atual aresta (i,j) do topo da lista de economias
Passo 7.1: **função de verificar se nós i e j já estão em alguma outra rota**
Passo 7.2: função para verificar se as restrições estão sendo atendidas.
Passo 7.3: Se passou em 7.1 e 7.2, i e j passam a fazer parte da nova rota.

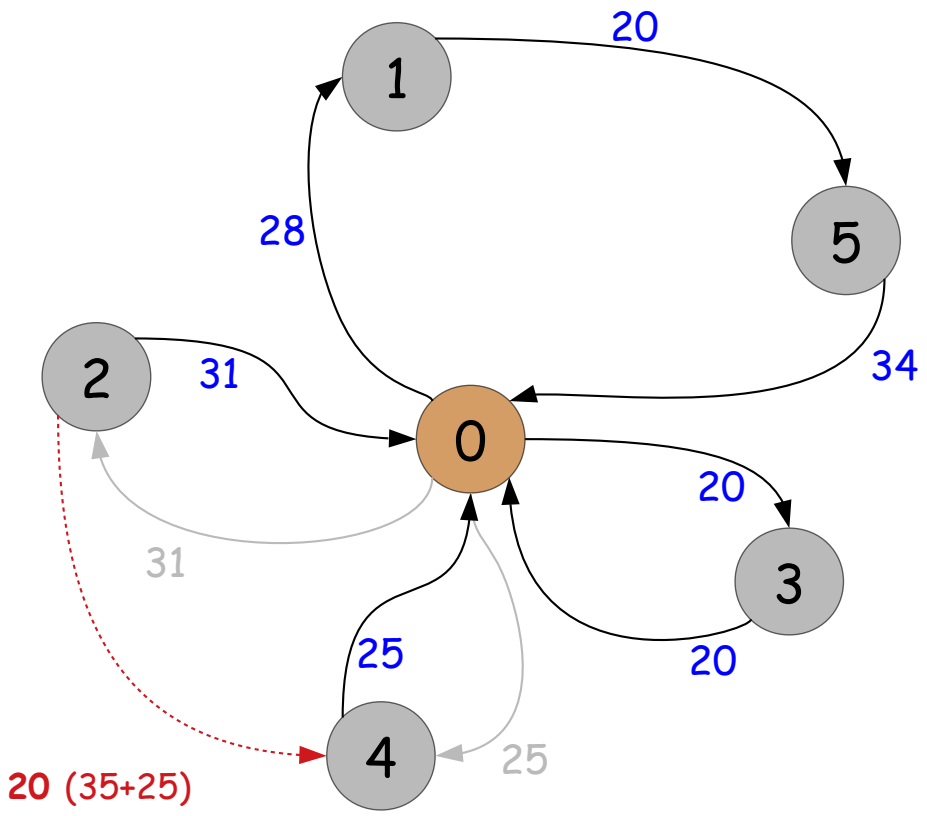


~~S_{15}~~	~~42~~
~~S_{12}~~	~~38~~
S_{24}	**36**
S_{45}	34
S_{25}	33
S_{14}	27
S_{35}	27
S_{13}	19
S_{34}	15
S_{23}	13

FALHOU!
Versão sequencial do algoritmo só permite a construção de uma rota de cada vez.
Como estamos construindo 0-1-5-0, não podemos iniciar a construção da rota 0-2-4-0

Problema Roteamento de Veículos

Passo 7: Tome a atual aresta (i,j) do topo da lista de economias
Passo 7.1: função de verificar se nós i e j já estão em alguma outra rota
Passo 7.2: **função para verificar se as restrições estão sendo atendidas.**
Passo 7.3: Se passou em 7.1 e 7.2, i e j passam a fazer parte da nova rota.

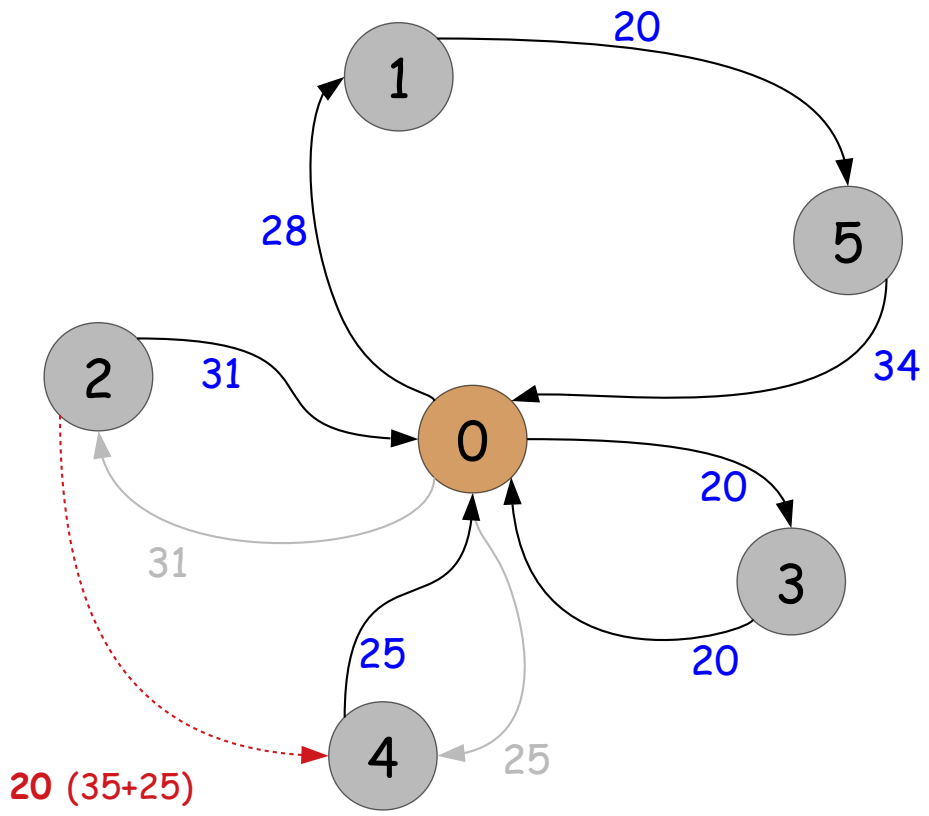


~~S_{15}~~	~~42~~
~~S_{12}~~	~~38~~
~~S_{24}~~	~~36~~
S_{45}	34
S_{25}	33
S_{14}	27
S_{35}	27
S_{13}	19
S_{34}	15
S_{23}	13

FALHOU!
Descarta aresta.

Problema Roteamento de Veículos

Passo 7: Tome a atual aresta (i,j) do topo da lista de economias
Passo 7.1: função de verificar se nós i e j já estão em alguma outra rota
Passo 7.2: função para verificar se as restrições estão sendo atendidas.
Passo 7.3: **Se passou em 7.1 e 7.2, i e j passam a fazer parte da nova rota.**



~~S_{15}~~	~~42~~
~~S_{12}~~	~~38~~
~~S_{24}~~	~~36~~
S_{45}	34
S_{25}	33
S_{14}	27
S_{35}	27
S_{13}	19
S_{34}	15
S_{23}	13

FALHOU!
Descarta aresta.

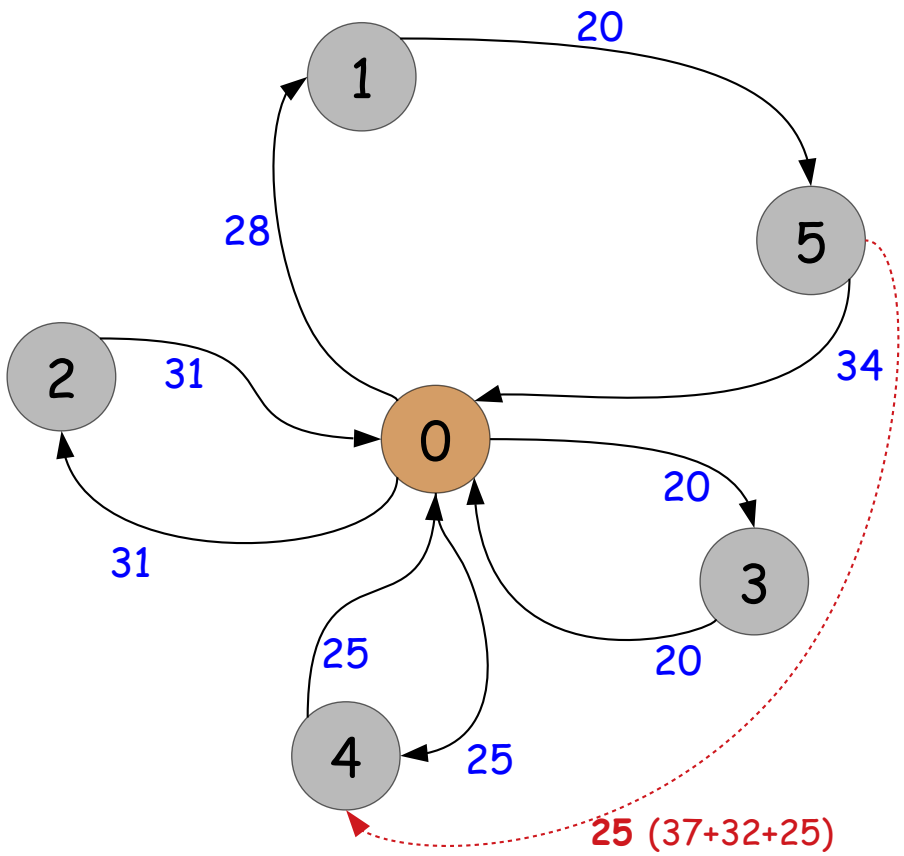
Problema Roteamento de Veículos

Passo 7: **Tome a atual aresta (i,j) do topo da lista de economias**

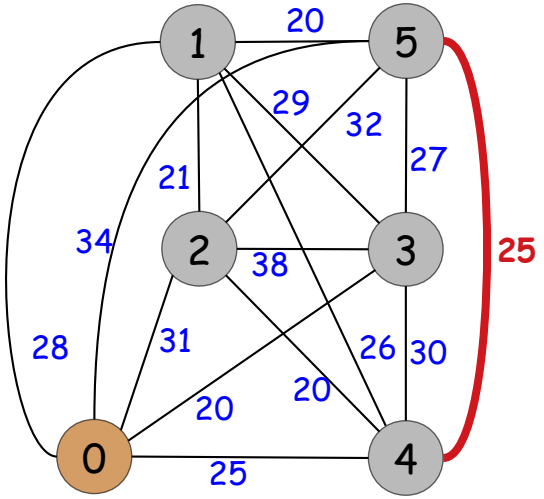
Passo 7.1: função de verificar se nós i e j já estão em alguma outra rota

Passo 7.2: função para verificar se as restrições estão sendo atendidas.

Passo 7.3: Se passou em 7.1 e 7.2, i e j passam a fazer parte da nova rota.



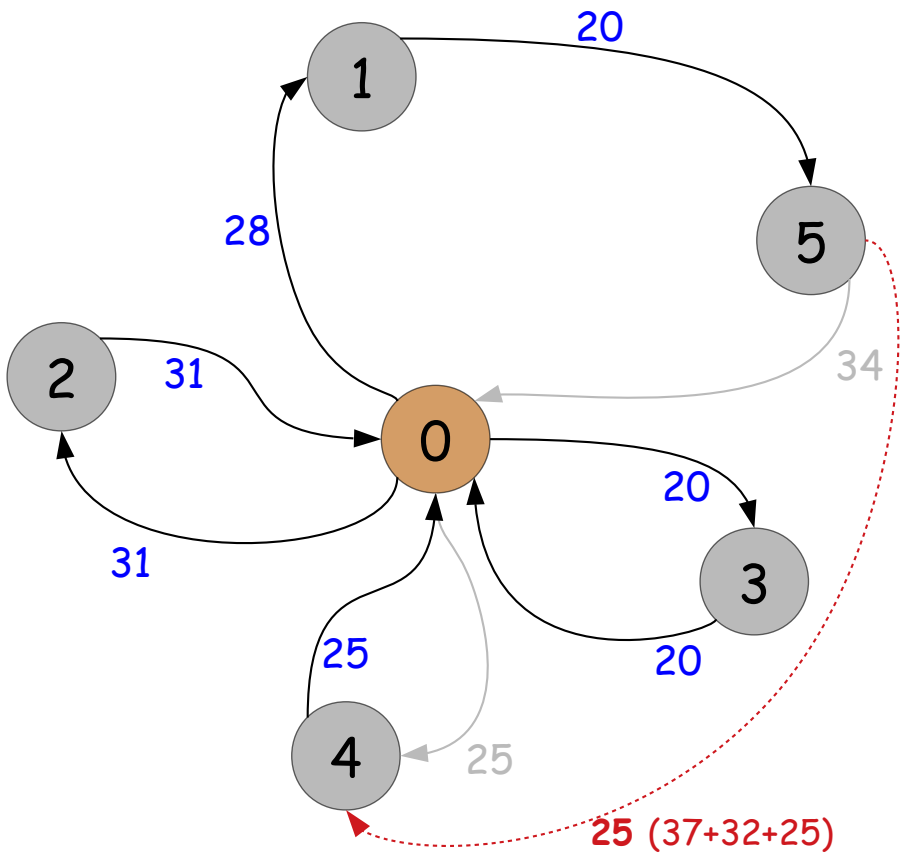
~~S_{15}~~	~~42~~
~~S_{12}~~	~~38~~
~~S_{24}~~	~~36~~
S_{45}	**34**
S_{25}	33
S_{14}	27
S_{35}	27
S_{13}	19
S_{34}	15
S_{23}	13



Demandas	
1	37
2	35
3	30
4	**25**
5	**32**

Problema Roteamento de Veículos

Passo 7: Tome a atual aresta (i,j) do topo da lista de economias
Passo 7.1: **função de verificar se nós i e j já estão em alguma outra rota**
Passo 7.2: função para verificar se as restrições estão sendo atendidas.
Passo 7.3: Se passou em 7.1 e 7.2, i e j passam a fazer parte da nova rota.

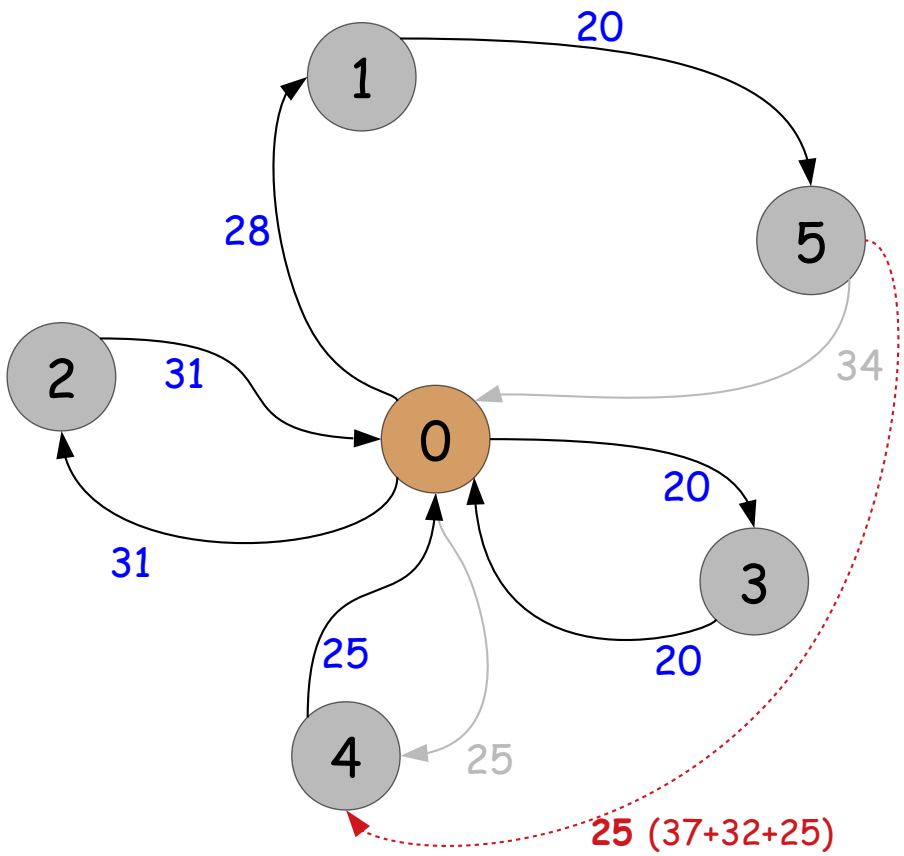


~~S_{15}~~	~~42~~
~~S_{12}~~	~~38~~
~~S_{24}~~	~~36~~
S_{45}	**34**
S_{25}	33
S_{14}	27
S_{35}	27
S_{13}	19
S_{34}	15
S_{23}	13

OK! Rotas podem ser unidadas

Problema Roteamento de Veículos

Passo 7: Tome a atual aresta (i,j) do topo da lista de economias
Passo 7.1: função de verificar se nós i e j já estão em alguma outra rota
Passo 7.2: **função para verificar se as restrições estão sendo atendidas.**
Passo 7.3: Se passou em 7.1 e 7.2, i e j passam a fazer parte da nova rota.

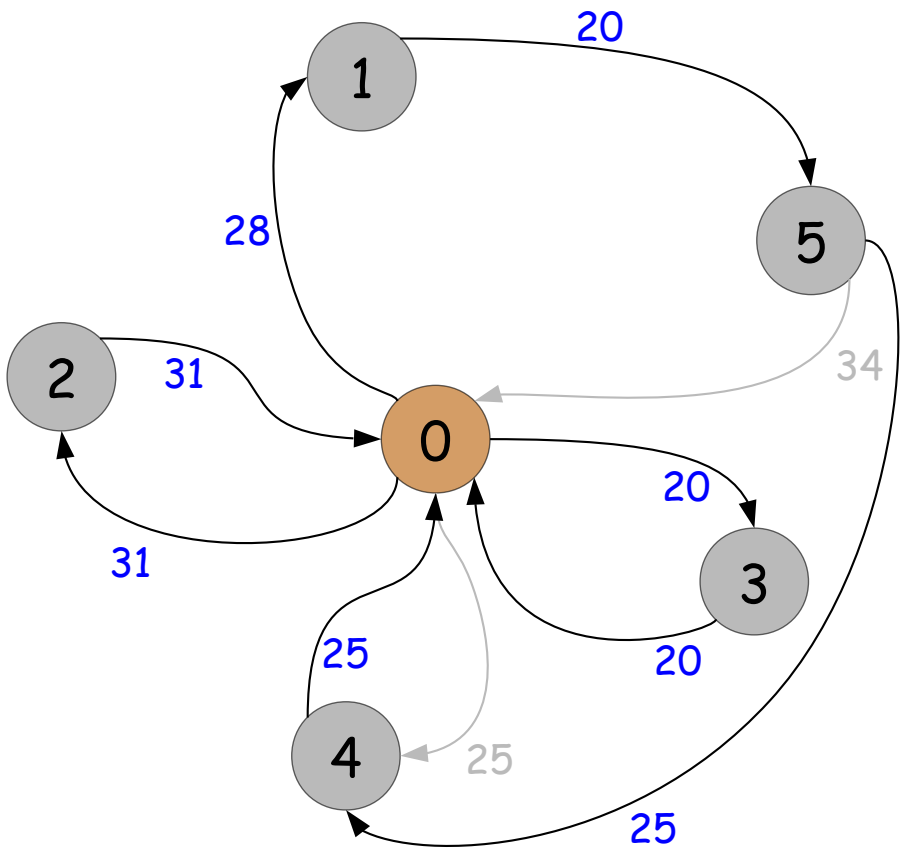


~~S_{15}~~	~~42~~
~~S_{12}~~	~~38~~
~~S_{24}~~	~~36~~
S_{45}	**34**
S_{25}	33
S_{14}	27
S_{35}	27
S_{13}	19
S_{34}	15
S_{23}	13

OK!
Única restrição é limite do veículo = 100
Demanda = 37+32 +25
Demanda = 98 < 100

Problema Roteamento de Veículos

Passo 7: Tome a atual aresta (i,j) do topo da lista de economias
Passo 7.1: função de verificar se nós i e j já estão em alguma outra rota
Passo 7.2: função para verificar se as restrições estão sendo atendidas.
Passo 7.3: **Se passou em 7.1 e 7.2, i e j passam a fazer parte da nova rota.**

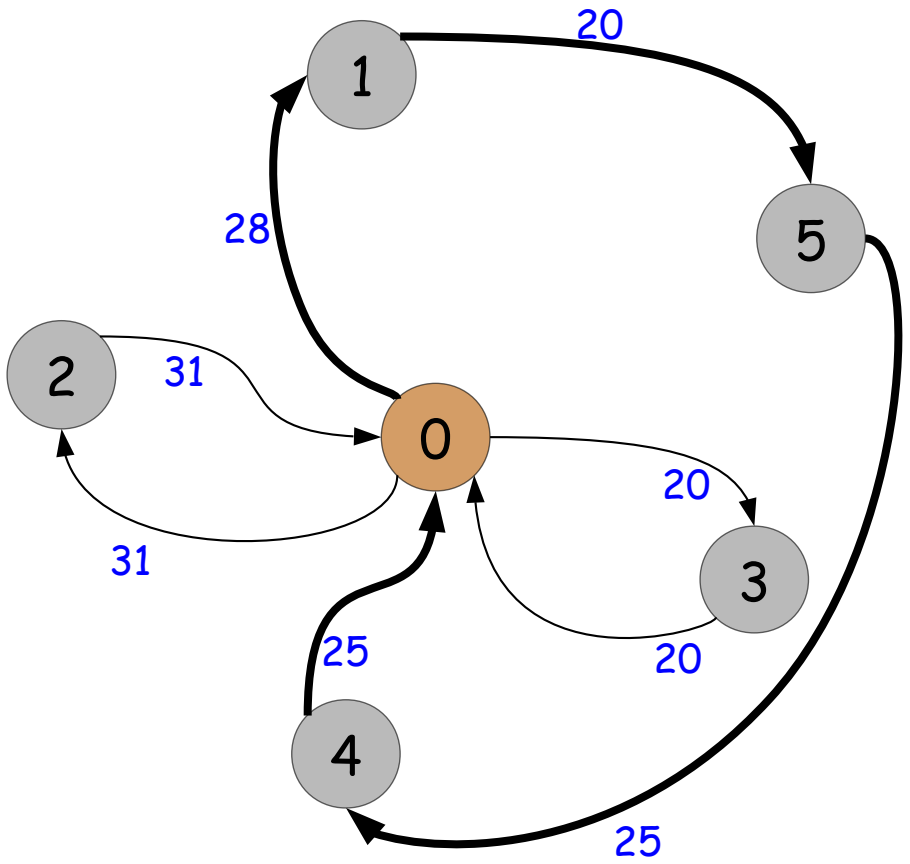


~~S_{15}~~	~~42~~
~~S_{12}~~	~~38~~
~~S_{24}~~	~~36~~
S_{45}	**34**
S_{25}	33
S_{14}	27
S_{35}	27
S_{13}	19
S_{34}	15
S_{23}	13

São necessários agora 3 veículos ao custo total de
276-42 = 234
234-34 = 200

Problema Roteamento de Veículos

Passo 7: Tome a atual aresta (i,j) do topo da lista de economias
Passo 7.1: função de verificar se nós i e j já estão em alguma outra rota
Passo 7.2: função para verificar se as restrições estão sendo atendidas.
Passo 7.3: **Se passou em 7.1 e 7.2, i e j passam a fazer parte da nova rota.**

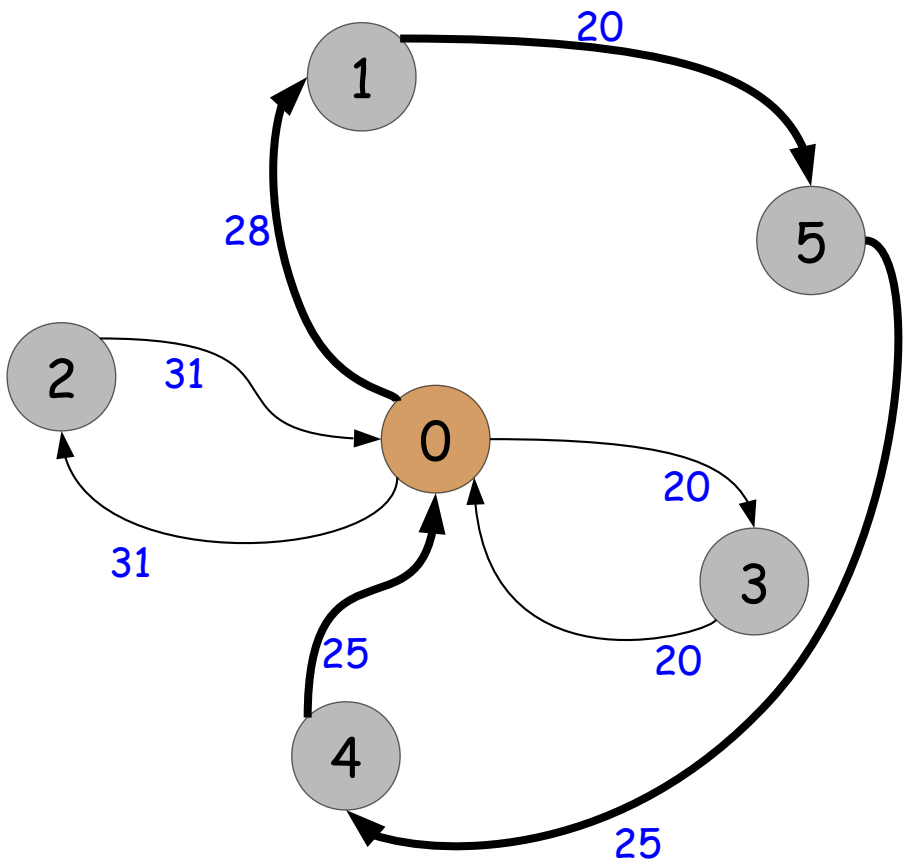


~~S_{15}~~	~~42~~
~~S_{12}~~	~~38~~
~~S_{24}~~	~~36~~
S_{45}	**34**
S_{25}	33
S_{14}	27
S_{35}	27
S_{13}	19
S_{34}	15
S_{23}	13

São necessários agora 3 veículos ao custo total de
276-42 = 234
234-34 = 200

Problema Roteamento de Veículos

Passo 7: **Tome a atual aresta (i,j) do topo da lista de economias**
Passo 7.1: função de verificar se nós i e j já estão em alguma outra rota
Passo 7.2: função para verificar se as restrições estão sendo atendidas.
Passo 7.3: Se passou em 7.1 e 7.2, i e j passam a fazer parte da nova rota.



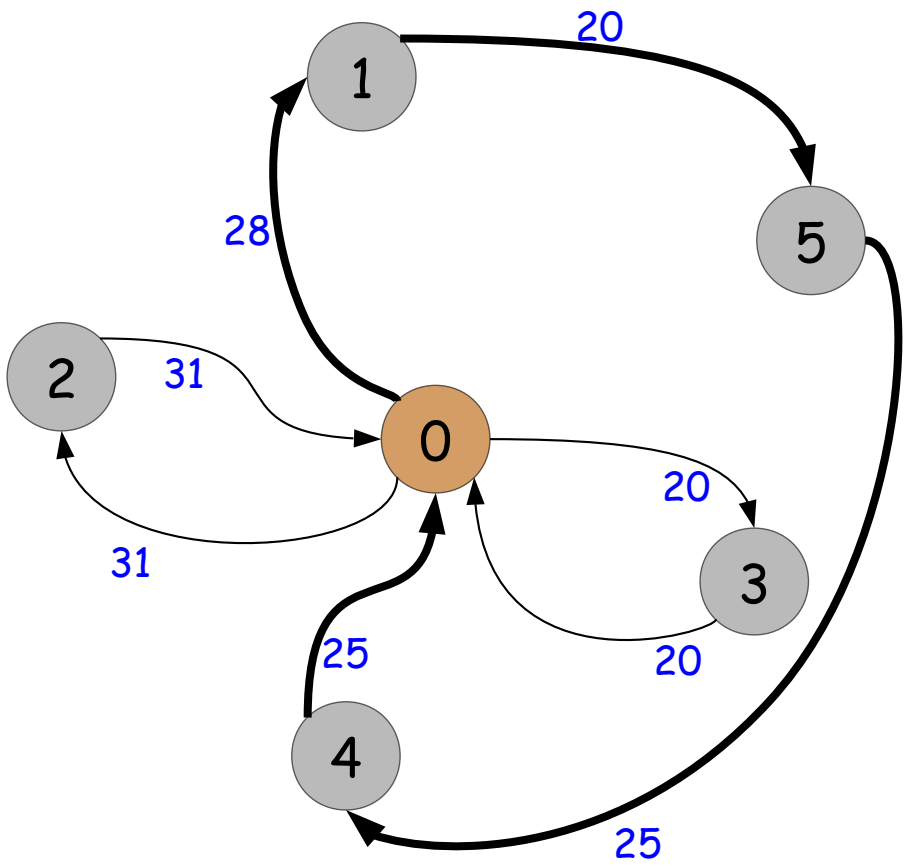
~~S_{15}~~	~~42~~
~~S_{12}~~	~~38~~
~~S_{24}~~	~~36~~
~~S_{45}~~	~~34~~
S_{25}	33
S_{14}	27
S_{35}	27
S_{13}	19
S_{34}	15
S_{23}	13

Na versão sequencial, todas as próximas arestas falharão no **Passo 7.2** já que a rota em construção, 0-1-5-4-0 possui demanda de 98.
Limite do veículo é 100.

Demandas	
1	37
2	35
3	30
4	25
5	32

Problema Roteamento de Veículos

Passo 7: **Tome a atual aresta (i,j) do topo da lista de economias**
Passo 7.1: função de verificar se nós i e j já estão em alguma outra rota
Passo 7.2: função para verificar se as restrições estão sendo atendidas.
Passo 7.3: Se passou em 7.1 e 7.2, i e j passam a fazer parte da nova rota.



~~S_{15}~~	~~42~~
~~S_{12}~~	~~38~~
~~S_{24}~~	~~36~~
~~S_{45}~~	~~34~~
~~S_{25}~~	~~33~~
~~S_{14}~~	~~27~~
~~S_{35}~~	~~27~~
~~S_{13}~~	~~19~~
~~S_{34}~~	~~15~~
~~S_{23}~~	~~13~~

Na versão sequencial, iniciamos a contrução de uma nova rota a partir do topo da lista.

Demandas	
1	37
2	35
3	30
4	25
5	32

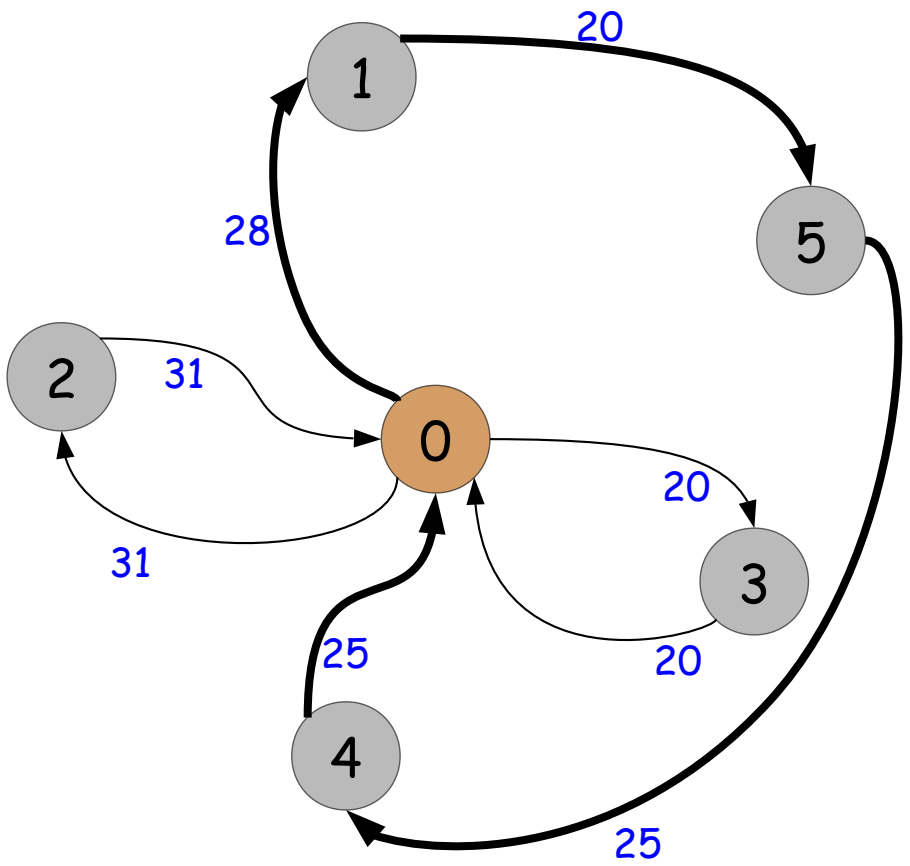
Problema Roteamento de Veículos

Passo 7: **Tome a atual aresta (i,j) do topo da lista de economias**

Passo 7.1: função de verificar se nós i e j já estão em alguma outra rota

Passo 7.2: função para verificar se as restrições estão sendo atendidas.

Passo 7.3: Se passou em 7.1 e 7.2, i e j passam a fazer parte da nova rota.



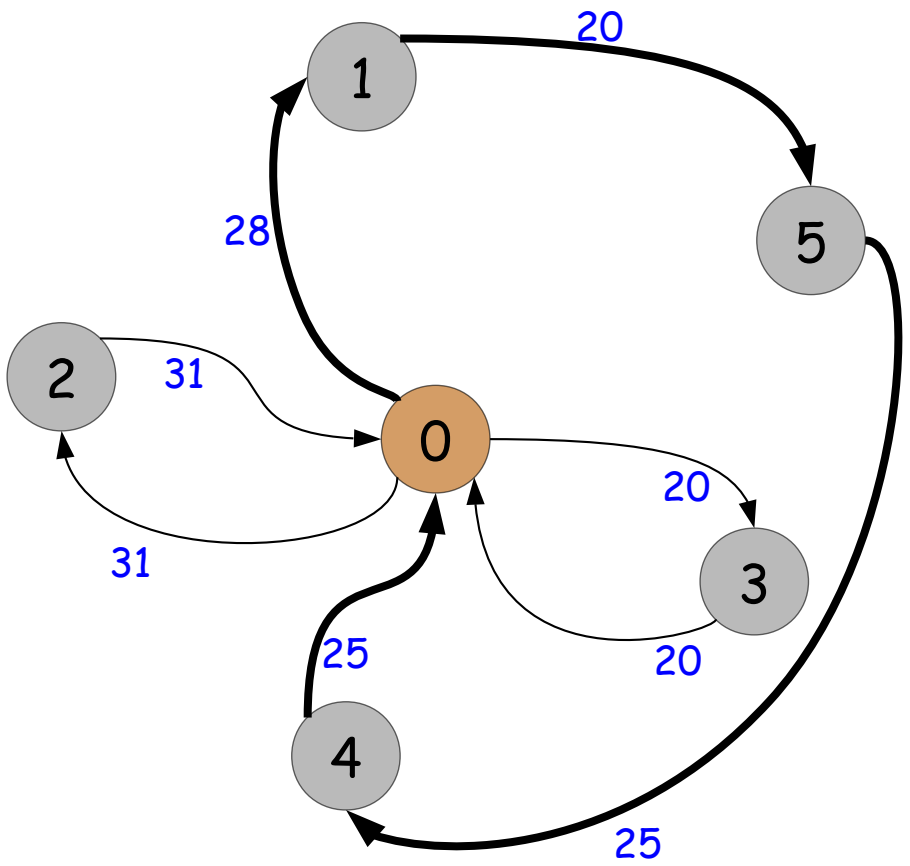
S_{15}	42
S_{12}	38
S_{24}	36
S_{45}	34
S_{25}	33
S_{14}	27
S_{35}	27
S_{13}	19
S_{34}	15
S_{23}	13

Na versão sequencial, iniciamos a contrução de uma nova rota a partir do topo da lista.

Demandas	
1	37
2	35
3	30
4	25
5	32

Problema Roteamento de Veículos

Passo 7: **Tome a atual aresta (i,j) do topo da lista de economias**
Passo 7.1: função de verificar se nós i e j já estão em alguma outra rota
Passo 7.2: função para verificar se as restrições estão sendo atendidas.
Passo 7.3: Se passou em 7.1 e 7.2, i e j passam a fazer parte da nova rota.



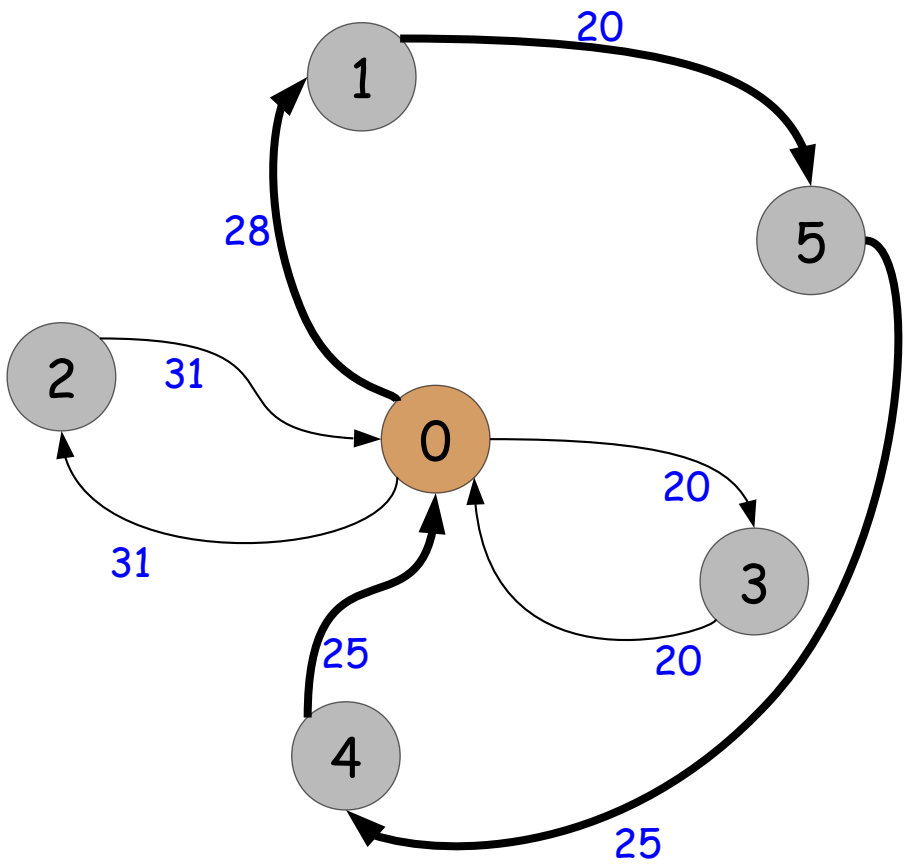
S_{15}	42
S_{12}	38
S_{24}	36
S_{45}	34
S_{25}	33
S_{14}	27
S_{35}	27
S_{13}	19
S_{34}	15
S_{23}	13

As nove primeiras arestas falharão no **Passo 7.1** pois possuem um nó na rota já construída.

Demandas	
1	37
2	35
3	30
4	25
5	32

Problema Roteamento de Veículos

Passo 7: **Tome a atual aresta (i,j) do topo da lista de economias**
Passo 7.1: função de verificar se nós i e j já estão em alguma outra rota
Passo 7.2: função para verificar se as restrições estão sendo atendidas.
Passo 7.3: Se passou em 7.1 e 7.2, i e j passam a fazer parte da nova rota.



~~S_{15}~~	~~42~~
~~S_{12}~~	~~38~~
~~S_{24}~~	~~36~~
~~S_{45}~~	~~34~~
~~S_{25}~~	~~33~~
~~S_{14}~~	~~27~~
~~S_{35}~~	~~27~~
~~S_{13}~~	~~19~~
~~S_{34}~~	~~15~~
S_{23}	13

As nove primeiras arestas falharão no **Passo 7.1** pois possuem um nó na rota já construída.

Demandas	
1	37
2	35
3	30
4	25
5	32

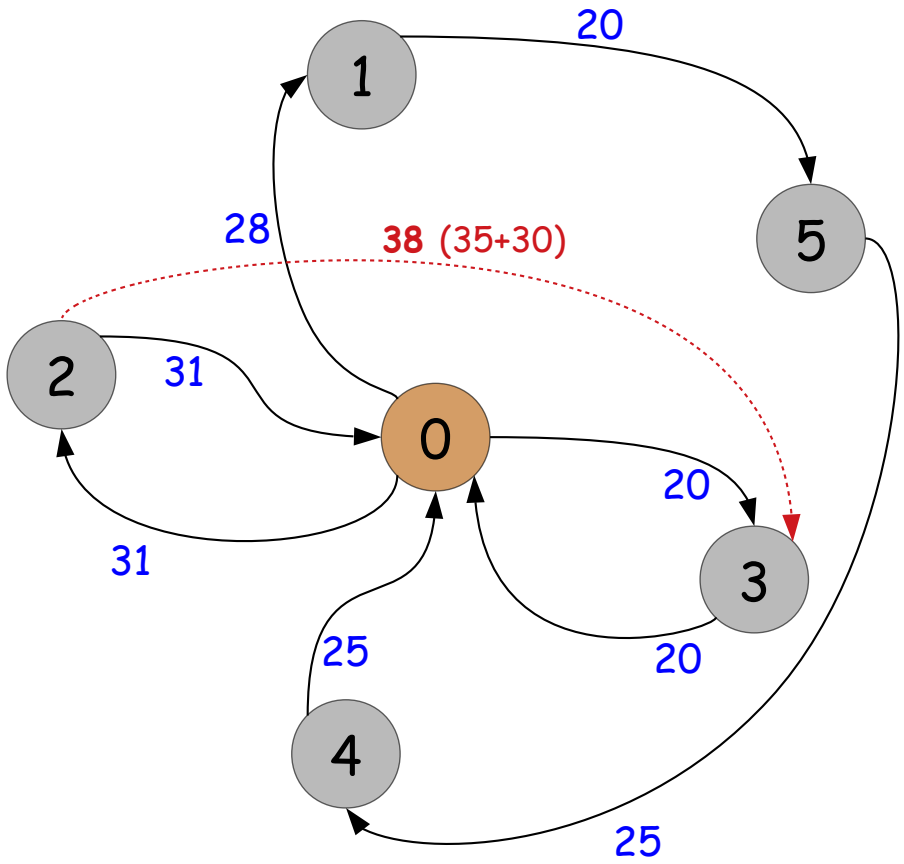
Problema Roteamento de Veículos

Passo 7: **Tome a atual aresta (i,j) do topo da lista de economias**

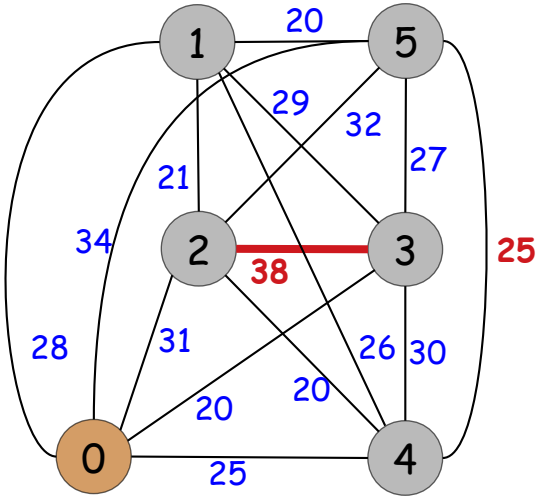
Passo 7.1: função de verificar se nós i e j já estão em alguma outra rota

Passo 7.2: função para verificar se as restrições estão sendo atendidas.

Passo 7.3: Se passou em 7.1 e 7.2, i e j passam a fazer parte da nova rota.



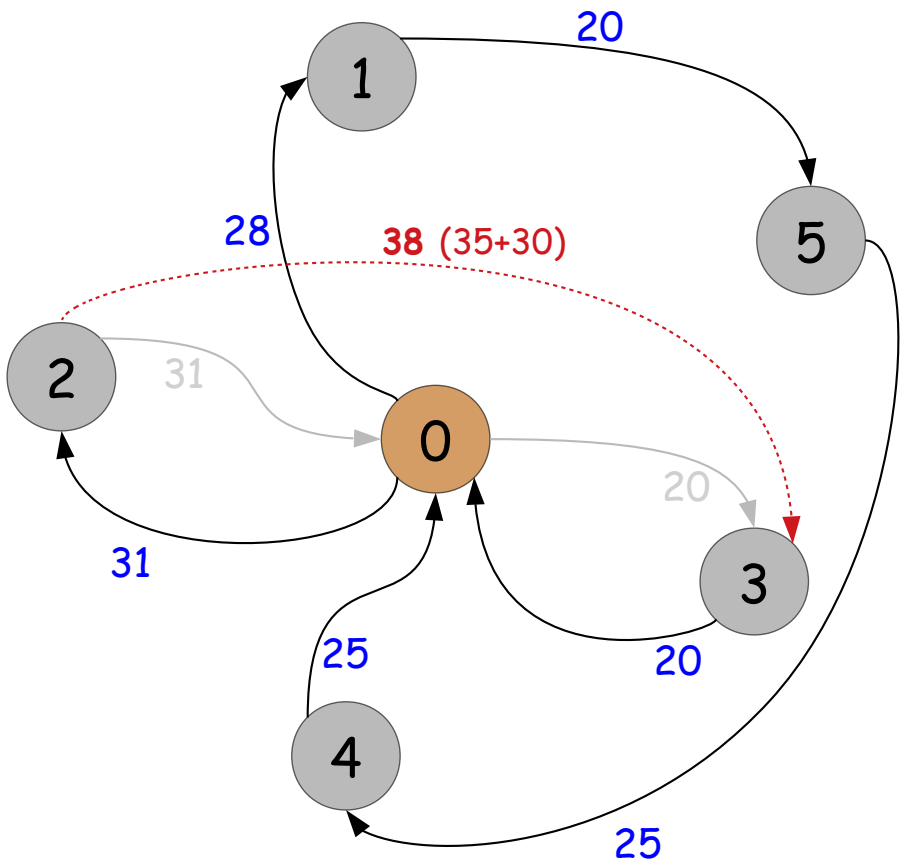
~~S_{15}~~	~~42~~
~~S_{12}~~	~~38~~
~~S_{24}~~	~~36~~
~~S_{45}~~	~~34~~
~~S_{25}~~	~~33~~
~~S_{14}~~	~~27~~
~~S_{35}~~	~~27~~
~~S_{13}~~	~~19~~
~~S_{34}~~	~~15~~
S_{23}	13



Demandas	
1	37
2	35
3	30
4	25
5	32

Problema Roteamento de Veículos

Passo 7: Tome a atual aresta (i,j) do topo da lista de economias
Passo 7.1: **função de verificar se nós i e j já estão em alguma outra rota**
Passo 7.2: função para verificar se as restrições estão sendo atendidas.
Passo 7.3: Se passou em 7.1 e 7.2, i e j passam a fazer parte da nova rota.

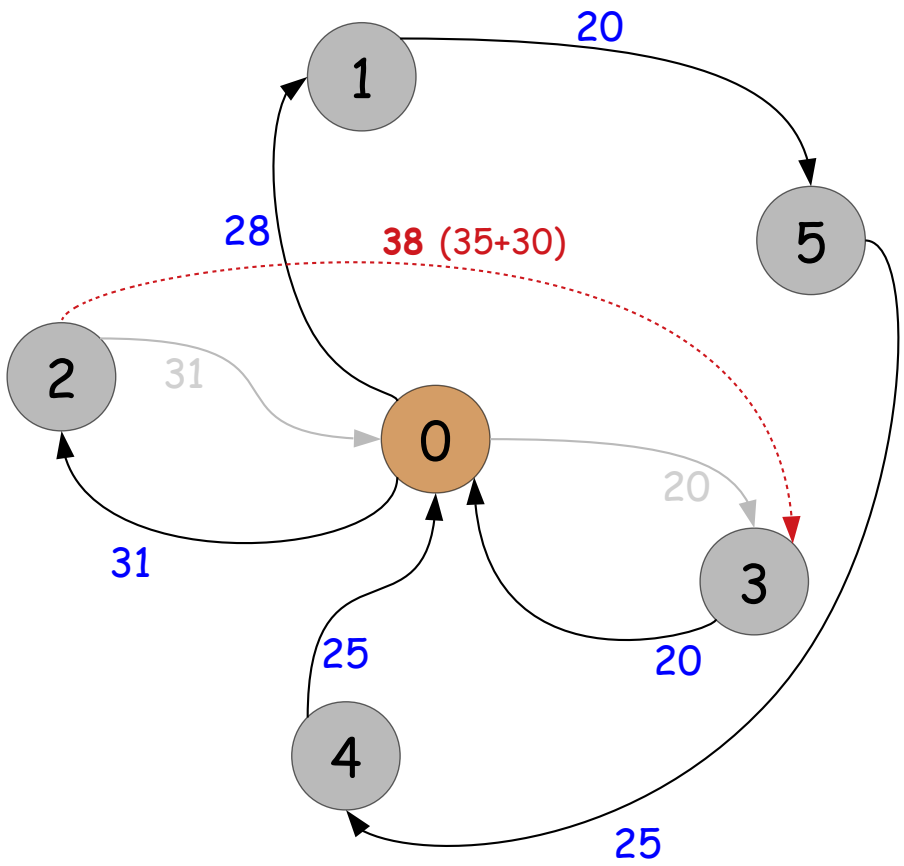


~~S_{15}~~	~~42~~
~~S_{12}~~	~~38~~
~~S_{24}~~	~~36~~
~~S_{45}~~	~~34~~
~~S_{25}~~	~~33~~
~~S_{14}~~	~~27~~
~~S_{35}~~	~~27~~
~~S_{13}~~	~~19~~
~~S_{34}~~	~~15~~
S_{23}	13

OK! Rotas podem ser unidadas

Problema Roteamento de Veículos

Passo 7: Tome a atual aresta (i,j) do topo da lista de economias
Passo 7.1: função de verificar se nós i e j já estão em alguma outra rota
Passo 7.2: **função para verificar se as restrições estão sendo atendidas.**
Passo 7.3: Se passou em 7.1 e 7.2, i e j passam a fazer parte da nova rota.

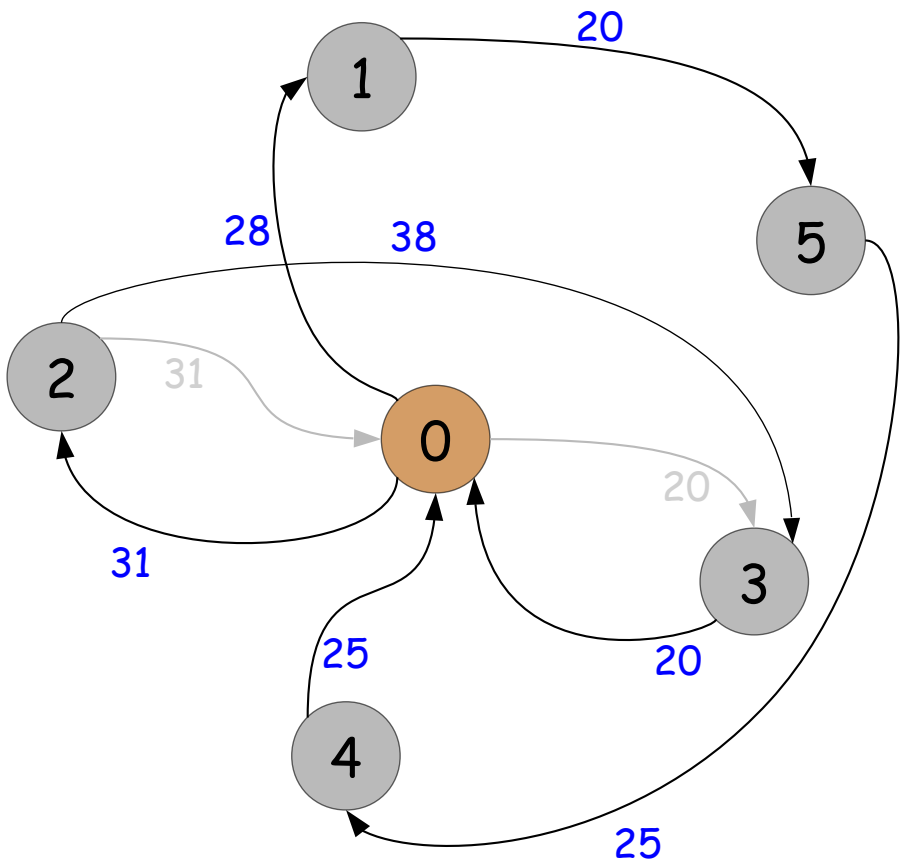


~~S_{15}~~	~~42~~
~~S_{12}~~	~~38~~
~~S_{24}~~	~~36~~
~~S_{45}~~	~~34~~
~~S_{25}~~	~~33~~
~~S_{14}~~	~~27~~
~~S_{35}~~	~~27~~
~~S_{13}~~	~~19~~
~~S_{34}~~	~~15~~
S_{23}	13

OK!
Única restrição é limite do veículo = 100
Demanda = 35+30
Demanda = 75 < 100

Problema Roteamento de Veículos

Passo 7: Tome a atual aresta (i,j) do topo da lista de economias
Passo 7.1: função de verificar se nós i e j já estão em alguma outra rota
Passo 7.2: função para verificar se as restrições estão sendo atendidas.
Passo 7.3: **Se passou em 7.1 e 7.2, i e j passam a fazer parte da nova rota.**

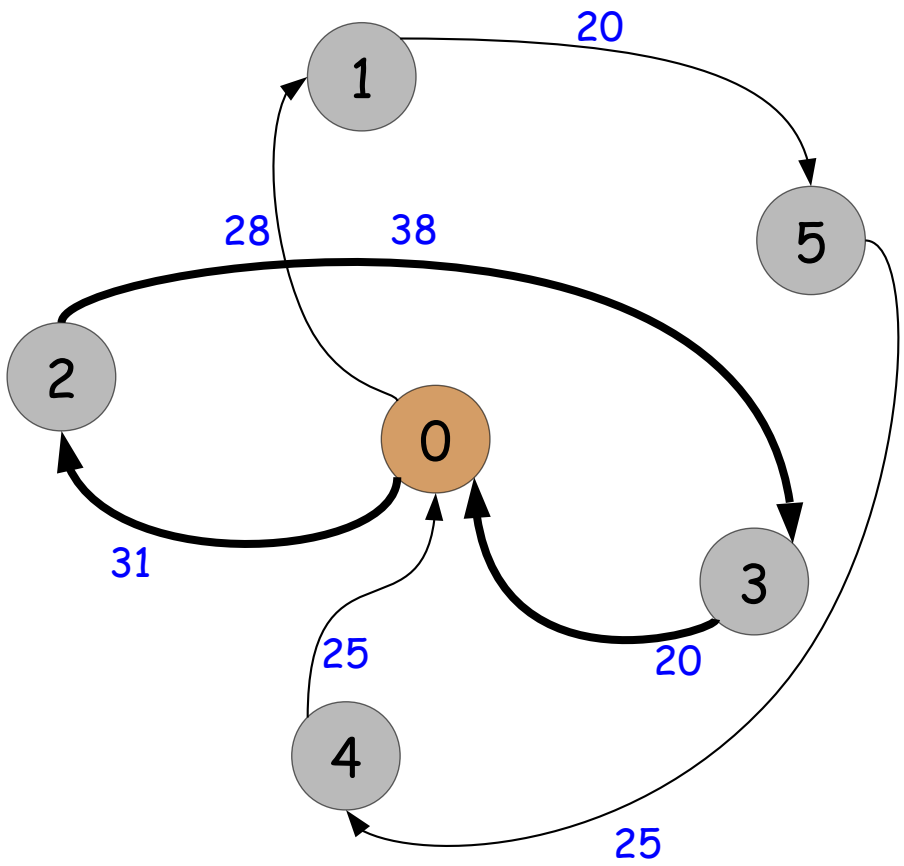


~~S_{15}~~	~~42~~
~~S_{12}~~	~~38~~
~~S_{24}~~	~~36~~
~~S_{45}~~	~~34~~
~~S_{25}~~	~~33~~
~~S_{14}~~	~~27~~
~~S_{35}~~	~~27~~
~~S_{13}~~	~~19~~
~~S_{34}~~	~~15~~
S_{23}	13

São necessários agora 2 veículos ao custo total de
276-42 = 234
234-34 = 200
200-13 = 187

Problema Roteamento de Veículos

Passo 7: Tome a atual aresta (i,j) do topo da lista de economias
Passo 7.1: função de verificar se nós i e j já estão em alguma outra rota
Passo 7.2: função para verificar se as restrições estão sendo atendidas.
Passo 7.3: **Se passou em 7.1 e 7.2, i e j passam a fazer parte da nova rota.**

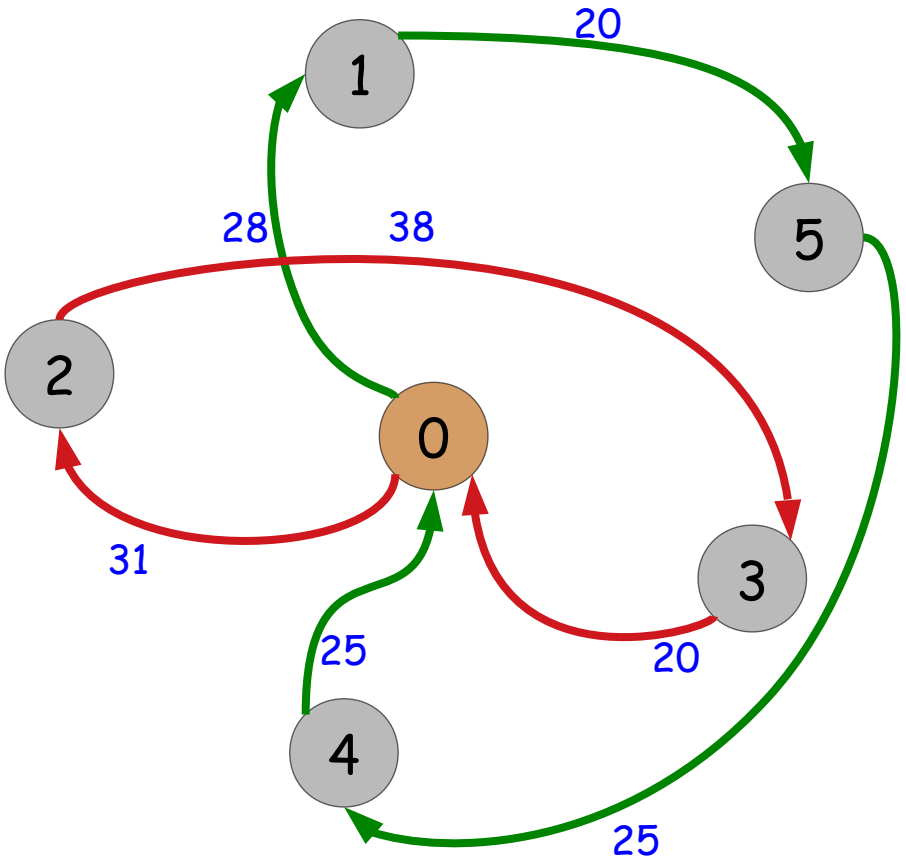


~~S_{15}~~	~~42~~
~~S_{12}~~	~~38~~
~~S_{24}~~	~~36~~
~~S_{45}~~	~~34~~
~~S_{25}~~	~~33~~
~~S_{14}~~	~~27~~
~~S_{35}~~	~~27~~
~~S_{13}~~	~~19~~
~~S_{34}~~	~~15~~
~~S_{23}~~	~~13~~

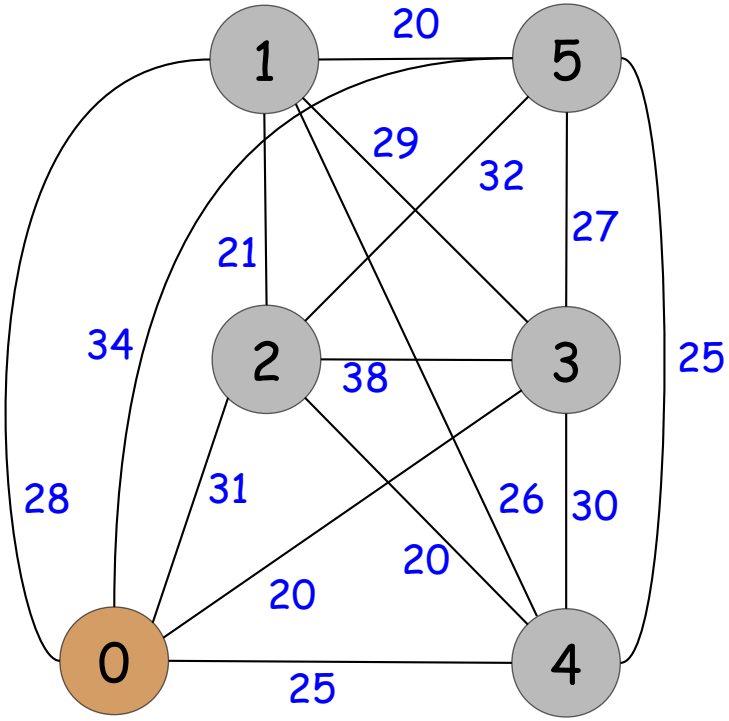
São necessários agora 2 veículos ao custo total de
276-42 = 234
234-34 = 200
200-13 = 187

FIM!

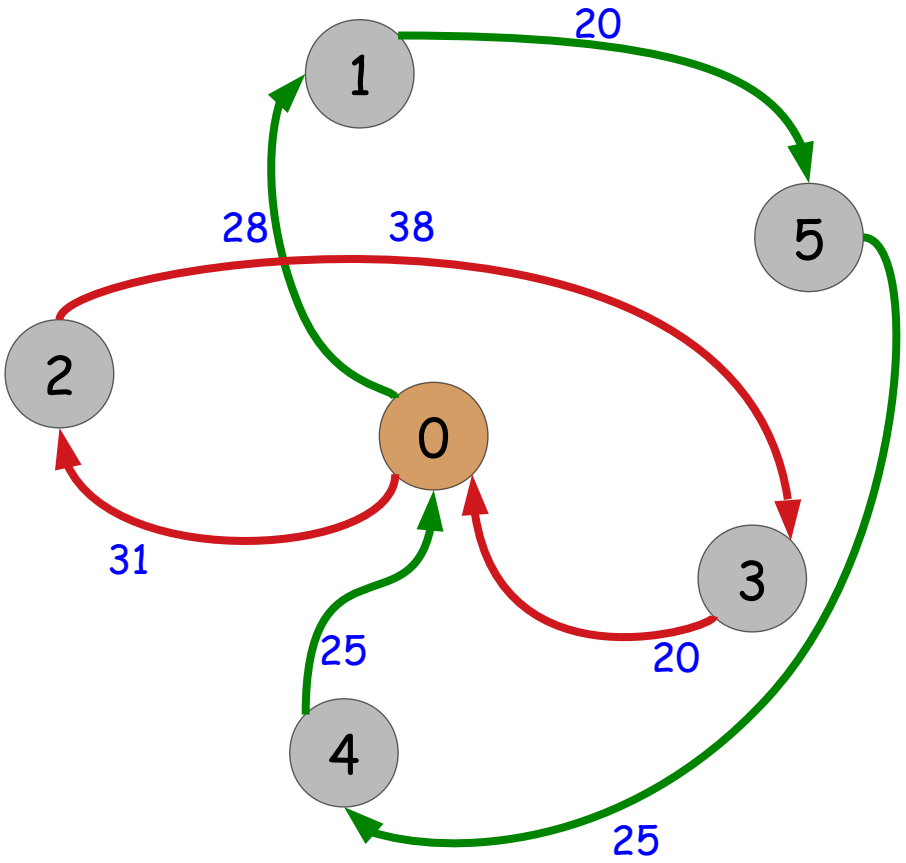
Problema Roteamento de Veículos



Entrada: (instância)



Problema Roteamento de Veículos



Saída: (solução)