

Pickup e Delivery com Janelas de Tempo: um Programa Inteiro

Arthur Henrique Sousa Cruz

Departamento de Ciência da Computação
Universidade Federal de Lavras

Programação Linear e Inteira



Sumário

- 1 Introdução
- 2 Programa Inteiro
- 3 Feasibility Pump
- 4 Testes e Resultados
- 5 Conclusão
- 6 Referências

- 1 Introdução
- 2 Programa Inteiro
- 3 Feasibility Pump
- 4 Testes e Resultados
- 5 Conclusão
- 6 Referências

Pickup and Delivery com Janelas de Tempo

- Variação do Problema de Roteamento de Veículos
- Pedidos: *pickups* e *deliveries*
- Janelas de Tempo
- Tempo de Serviço

Pickup and Delivery com Janelas de Tempo

- NP-difícil
- Modelo adaptado de Ropke and Pisinger (2006)
- Feasibility Pump Fischetti et al. (2005)

Considerações

- $a_i \geq 0$ e $b_i \geq 0$
- $c_{ij} \geq 0$
- $t_{ij} \geq 0$
- $s_i \geq 0$
- $t_{ij} + s_i > 0$
- $t_{ij} + t_{jk} \geq t_{ik}$

- 1 Introdução
- 2 Programa Inteiro**
- 3 Feasibility Pump
- 4 Testes e Resultados
- 5 Conclusão
- 6 Referências

Variáveis

- $x_{ijk} \in \{0, 1\}$: igual a 1 se a aresta $(i, j) \in A$ está na rota do veículo k
- $z_i \in \{0, 1\}$: igual a 1 se o pedido referente ao *pickup* i não é atendido
- $S_{ik} \in \mathbb{N}$: indica quando o veículo k iniciou o serviço no vértice i
- $L_{ik} \in \mathbb{N}$: *upper bound* para a carga do veículo k após o servir no vértice i

Função Objetivo

$$\min \alpha \sum_{k \in K} \sum_{(i,j) \in A} d_{ij} x_{ijk} + \beta \sum_{k \in K} S_{\sigma k} - a_{\tau} + \gamma \sum_{i \in P} z_i \quad (1)$$

Restrição Pedidos Realizados

$$\sum_{k \in K} \sum_{j \in N} x_{ijk} + z_i = 1, \forall i \in P \quad (2)$$

Delivery \leftrightarrow Pickup

$$\sum_{j \in V} x_{ijk} - \sum_{j \in V} x_{jmk} = 0, \forall k \in K, \forall i \in P, m = \text{par}(i) \quad (3)$$

Terminal Inicial, Terminal Final e Continuidade

$$\sum_{j \in P \cup \{\sigma\}} x_{\tau j k} = 1, \forall k \in K \quad (4)$$

$$\sum_{i \in D \cup \{\tau\}} x_{i \sigma k} = 1, \forall k \in K \quad (5)$$

$$\sum_{i \in V} x_{ijk} - \sum_{i \in V} x_{jik} = 0, \forall k \in K, \forall j \in N \quad (6)$$

Tempo de início de serviço e Subciclos

$$S_{ik} + s_i + t_{ij} \leq (1 - x_{ijk})M + S_{jk}, \forall k \in K, \forall (i, j) \in A \quad (7)$$

$$a_i \leq S_{ik} \leq b_i, \forall k \in K, \forall i \in V \quad (8)$$

Pickups antes de Deliveries

$$S_{ik} \leq S_{jk}, \forall k \in K, \forall i \in P, j = \text{par}(i) \quad (9)$$

Upper Bound da capacidade

$$L_{ik} + l_i \leq (1 - x_{ijk})M + L_{jk}, \forall k \in K, \forall (i, j) \in A \quad (10)$$

$$L_{ik} \leq C, \forall k \in K, \forall i \in V \quad (11)$$

$$L_{\tau k} = L_{\sigma k} = 0, \forall k \in K \quad (12)$$

Domínio das variáveis

$$x_{ijk} \in \{0, 1\}, \forall k \in K, \forall (i, j) \in A \quad (13)$$

$$z_i \in \{0, 1\}, \forall i \in P \quad (14)$$

$$S_{ik} \geq 0, \forall k \in K, \forall i \in V \quad (15)$$

$$L_{ik} \geq 0, \forall k \in K, \forall i \in V \quad (16)$$

- 1 Introdução
- 2 Programa Inteiro
- 3 Feasibility Pump**
- 4 Testes e Resultados
- 5 Conclusão
- 6 Referências

Feasibility Pump

Algorithm 1 FP para o PDTJ

```
1:  $B^* = \text{resolve\_relaxacao\_linear}()$ 
2:  $\bar{B} = \text{arredonda}(B^*)$ 
3: while  $\text{eh\_otimo}(B^*)$  e  $\text{nao\_factive}$  e  $\text{nao\_estourou\_tempo}()$  do
4:    $B^* = \text{resolve\_nova\_relaxacao}(\bar{B})$ 
5:   if  $\text{arredonda}(B^*) \neq \bar{B}$  then
6:      $\bar{B} = \text{arredonda}(B^*)$ 
7:     if  $B^* == \bar{B}$  then
8:        $\text{nao\_factive} = \text{Falso}$ 
9:     end if
10:  else
11:     $\text{pertubacao}(\bar{B})$ 
12:  end if
13: end while
14: if  $\text{nao\_factive}$  then
15:   return  $\emptyset$ 
16: end if
17: return  $\bar{B}$ 
```

Nova Função Objetivo

$$\sum_{b \in \overline{B}_{bin} | x=0} b + \sum_{b \in \overline{B}_{bin} | b=1} 1 - b + \delta \sum_{b \in \overline{B}_S} b - \rho \sum_{b \in \overline{B}_L} b \quad (17)$$

Soluções Triviais

$$x_{\tau\sigma k} = 0 \quad (18)$$

$$x_{\sigma\tau k} = 0 \quad (19)$$

- 1 Introdução
- 2 Programa Inteiro
- 3 Feasibility Pump
- 4 Testes e Resultados**
- 5 Conclusão
- 6 Referências

Implementação e Instâncias

- Python3 + Gurobi
- 8 instâncias de Sartori and Buriol (2020)
- 50 pedidos cada

Implementação e Instâncias

- Terminal Inicial duplicado
- 5 veículos
- 25 minutos de execução

Resultaods

- Nenhuma solução factível
- Provável Motivo: As alterações no método

- 1 Introdução
- 2 Programa Inteiro
- 3 Feasibility Pump
- 4 Testes e Resultados
- 5 Conclusão**
- 6 Referências

Propôs-se para o PDJT

- Um Programa Inteiro
- Uma heurística baseada no Feasibility Pump

Resultados dos Experimentos

- Nenhuma solução factível
- Motivo: Variações aplicadas à heurística

Melhorias

- Implementação do método clássico
- Análise das restrições para soluções triviais

- 1 Introdução
- 2 Programa Inteiro
- 3 Feasibility Pump
- 4 Testes e Resultados
- 5 Conclusão
- 6 Referências**

- Matteo Fischetti, Fred Glover, and Andrea Lodi. The feasibility pump. *Mathematical Programming*, 104(1):91–104, 2005.
- Stefan Ropke and David Pisinger. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation science*, 40(4):455–472, 2006.
- Carlo S Sartori and Luciana S Buriol. A study on the pickup and delivery problem with time windows: Matheuristics and new instances. *Computers & Operations Research*, page 105065, 2020.

Pickup e Delivery com Janelas de Tempo: um Programa Inteiro

Arthur Henrique Sousa Cruz

Departamento de Ciência da Computação
Universidade Federal de Lavras

Programação Linear e Inteira

