

# Modelos e Algoritmos para Problemas Integrados de Distribuição e Roteamento

**Fernando Afonso Santos**

Orientador: Geraldo Robson Mateus

Co-orientador: Alexandre Salles da Cunha

**Defesa de Projeto de Tese**

Programa de Pós-Graduação em Ciência da Computação

Universidade Federal de Minas Gerais

# Sumário

- 1 **Introdução**
- 2 VRPCD
- 3 PDPCD
- 4 2EVRP
- 5 Cronograma

# Introdução

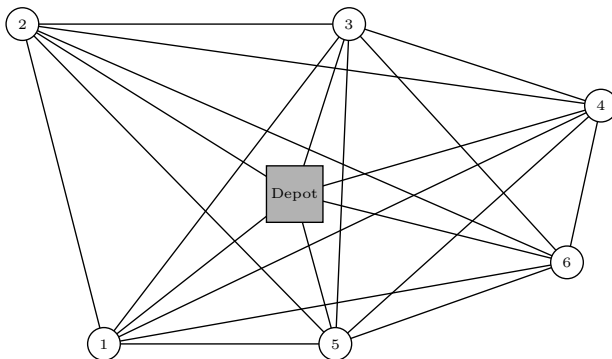
- O Problema de Roteamento de Veículos (PRV) foi proposto por Dantzig e Ramser (1951)

## Definição do PRV

- $G = (V, A)$  é um grafo direcionado
  - $V = \{0, 1, \dots, n\}$
  - $A = \{(i, j) : i, j \in V; i \neq j\}$
- Cada vértice  $i \in V \setminus \{0\}$  está associado a uma demanda  $q_i > 0$
- Frota homogênea de  $K$  veículos com capacidade  $Q$
- Cada vértice deve ser visitado exatamente uma vez para atender sua demanda
- Ao associar custos  $c_{ij}, \forall (i, j) \in A$ , o PRV consiste em encontrar a solução de menor custo em  $G$

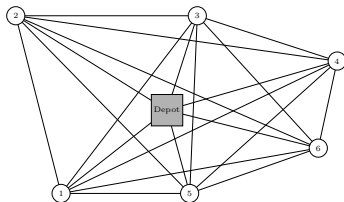
# Introdução

Exemplo de PRV:  $V = \{0, 1, 2, 3, 4, 5, 6\}$   
 $d_i = 1, \forall i \in V \setminus \{0\}$   
 $Q = 4, K = 2$



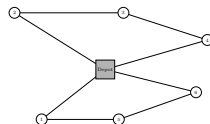
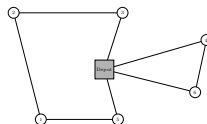
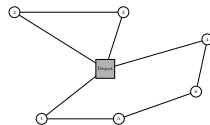
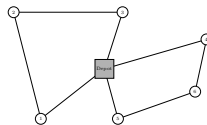
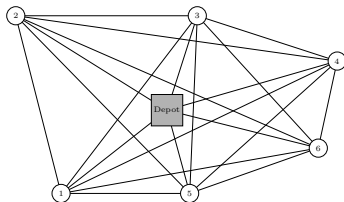
# Introdução

Exemplo de PRV:  $V = \{0, 1, 2, 3, 4, 5, 6\}$   
 $d_i = 1, \forall i \in V \setminus \{0\}$   
 $Q = 4, K = 2$



# Introdução

Exemplo de PRV:  $V = \{0, 1, 2, 3, 4, 5, 6\}$   
 $d_i = 1, \forall i \in V \setminus \{0\}$   
 $Q = 4, K = 2$



# Introdução

- Variações clássicas do PRV
  - PRV com Janelas de Tempo
  - PRV com Múltiplos Depósitos
  - PRV com Coleta e Entrega
- No PRV e suas variações clássicas, considera-se que o depósito funcione como uma garagem ou armazém
- Nos últimos anos empresas e operadores de logística têm proposto expandir a funcionalidade destes armazéns
- Em alguns contextos, os armazéns também podem servir para a consolidação de cargas, que consiste em organizá-las e prepará-las para a entrega
- Entre os armazéns com esta funcionalidade, destacam-se
  - *Cross-Dockings* (CD)
  - *Satellites*

# Introdução

- Recentemente, autores têm proposto novas variações do PRV que se adequem a estes novos cenários de roteamento
- Além das decisões de roteamento, são consideradas também decisões sobre a manipulação das mercadorias nos armazéns
- Entre os principais problemas se destacam
  - PRV com *Cross-Docking*  
(*Vehicle Routing Problem with Cross-Docking* - VRPCD)
  - PRV com Dois Elos  
(*Two-Echelon Vehicle Routing Problem* - 2EVRP)
- Neste projeto de tese, concentramos nossos estudos em modelos e algoritmos para a solução destes problemas



# Sumário

- 1 Introdução
- 2 VRPCD**
- 3 PDPCD
- 4 2EVRP
- 5 Cronograma

# Definições

## *Cross-Docking*

- Recebe mercadorias coletadas por um conjunto de veículos e permite que estas sejam consolidadas antes de serem encaminhadas para a entrega
- A consolidação pode ter motivação operacional ou econômica
  - **Operacional:** Pesagem das cargas, embalagem, gestão de cadeias de suprimentos, etc
  - **Econômica:** Arranjo das cargas nos veículos, de forma a minimizar os custos de roteamento
- Não possui estoque de longo prazo. As cargas podem permanecer no armazém por um curto período, no máximo 24h

# Definições

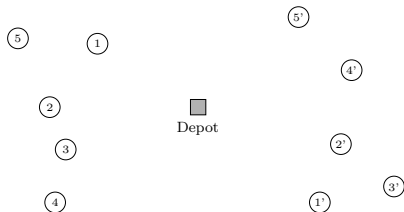
A integração do PRV com a consolidação de mercadorias no *Cross-Docking* dá origem a um novo problema: o Problema de Roteamento de Veículos com *Cross-Docking* (VRPCD)

## VRPCD

- $G = (\{S, C, 0\}, A_S \cup A_C)$ ,  $S = \{1, \dots, n\}$ ,  $C = \{1', \dots, n'\}$  e 0 são os fornecedores, consumidores e o *Cross-Docking*
- $A_S = \{(i, j) : i, j \in S \cup \{0\}\}$ ,  $A_C = \{(i, j) : i, j \in C \cup \{0\}\}$
- $n$  requisições dadas por  $P = \{p_i := (i, i', q_i) : i = 1, \dots, n\}$ : carga  $q_i > 0$  é coletada em  $i \in S$  e entregue em  $i' \in C$
- Uma frota homogênea de  $K$  veículos com capacidade  $Q$  é disponível no *Cross-Docking* para atender as requisições
- Dados custos  $c_{ij}$  nos arcos e  $c_i$  para trocar a carga  $q_i$  no CD, o VRPCD consiste em definir  $2K$  rotas e a estratégia de consolidação que minimize o custo total

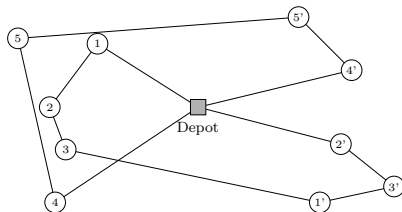
# Exemplo

## PRV com Coleta e Entrega



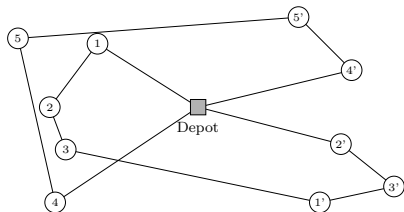
# Exemplo

## PRV com Coleta e Entrega

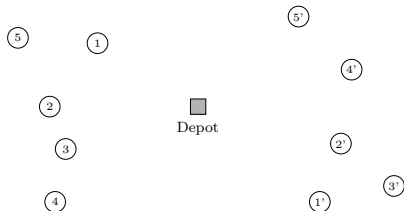


# Exemplo

## PRV com Coleta e Entrega

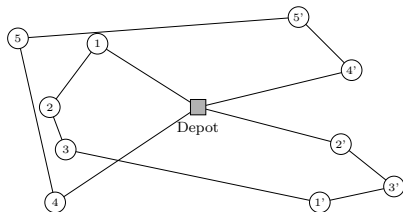


## PRV com *Cross-Docking*

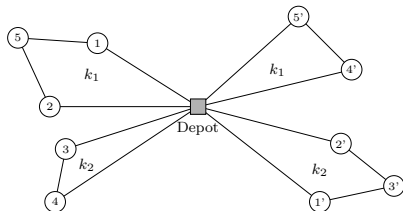


# Exemplo

PRV com Coleta e Entrega

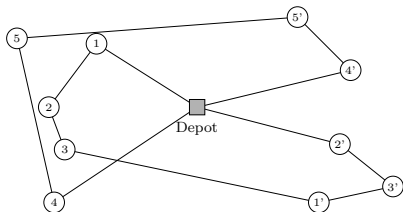


PRV com *Cross-Docking*

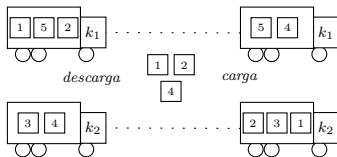
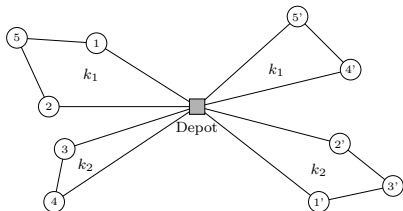


# Exemplo

## PRV com Coleta e Entrega



## PRV com *Cross-Docking*





# Trabalhos Relacionados

- O VRPCD foi introduzido por Lee et al. (2006)
  - Não foram considerados custos para a troca de mercadorias durante a consolidação
  - Problema com janelas de tempo tanto nos vértices quanto no *Cross-Docking*
- Wen et al. (2009) solucionaram um problema semelhante, considerando permitindo a consolidação assíncrona de mercadorias no *docking*
- Diversos trabalhos lidam apenas com o problema de consolidação no *docking*
  - Yu e Egbelu (2008)
  - Chen e Lee (2009)
  - Boysen et al (2010)

# Trabalhos Desenvolvido

- Foram propostas duas formulações matemáticas para solucionar o VRPCD
- Ambas as formulações apresentam um número exponencial de variáveis
- Para a solução das formulações foi utilizada a técnica de decomposição de geração de colunas e respectivos algoritmos Branch-and-price (BP)

# Formulação Matemática #1

- O modelo lida com rotas para fornecedores e consumidores de forma independente
- Os conjuntos  $R$  e  $R'$  denotam o conjunto de rotas viáveis para visitar fornecedores e consumidores

## Parâmetros

- $c_r(c_{r'})$ : custo de roteamento (soma dos custos dos arcos) da rota  $r \in R$  ( $r' \in R'$ )
- $a_{ir} \in \{0, 1\}$ : a rota  $r$  visita o vértice  $i$  ou não

## Variáveis de decisão

- $\lambda_r^k \in \{0, 1\}$ : a rota  $r \in R$  é percorrida ou não pelo veículo  $k$
- $\gamma_{r'}^k \in \{0, 1\}$ : a rota  $r' \in R'$  é percorrida ou não pelo veículo  $k$
- $\tau_i^k$ : a carga  $q_i$  é carregada/descarregada pelo veículo  $k$  no *Cross-Docking*

# Formulação Matemática #1

$$\min \quad \sum_{r \in R} c_r \sum_{k \in K} \lambda_r^k + \sum_{r' \in R'} c_{r'} \sum_{k \in K} \gamma_{r'}^k + \sum_{k \in K} \sum_{i \in S} c_i^k \tau_i^k \quad (1)$$

$$\sum_{r \in R} \lambda_r^k = 1 \quad k = 1, \dots, K \quad (2)$$

$$\sum_{r' \in R'} \gamma_{r'}^k = 1 \quad k = 1, \dots, K \quad (3)$$

$$\sum_{r \in R} a_{ir} \sum_{k=1}^K \lambda_r^k = 1 \quad \forall i \in S \quad (4)$$

$$\sum_{r' \in R'} a_{i'r'} \sum_{k=1}^K \gamma_{r'}^k = 1 \quad \forall i' \in C \quad (5)$$

$$\sum_{r \in R} \lambda_r^k a_{ir} - \sum_{r' \in R'} \gamma_{r'}^k a_{i'r'} + \tau_i^k \geq 0 \quad \forall p_i \in P, \quad k = 1, \dots, K \quad (6)$$

$$- \sum_{r \in R} \lambda_r^k a_{ir} + \sum_{r' \in R'} \gamma_{r'}^k a_{i'r'} + \tau_i^k \geq 0 \quad \forall p_i \in P, \quad k = 1, \dots, K \quad (7)$$

$$\lambda \in \mathbb{B}^{K|R|}, \gamma \in \mathbb{B}^{K|R'|}, \tau \in \mathbb{B}^{Kn}. \quad (8)$$

# Formulação Matemática #1

- A formulação (1)-(8) conta com um número exponencial de variáveis
- A enumeração explícita de todas as variáveis torna sua solução impraticável para problemas de médio ou grande porte
- Para a solução do modelo, foi utilizada a técnica de enumeração implícita de variáveis através da geração de colunas e branch-and-price

# Geração de Colunas #1

O Problema Linear Master (PLM) da Formulação #1 é obtido ao considerar a formulação (1)-(7) e

$$\lambda_r^k \geq 0 \quad \forall r \in R, k = 1, \dots, K \quad (9)$$

$$\gamma_{r'}^k \geq 0 \quad \forall r' \in R', k = 1, \dots, K \quad (10)$$

$$0 \leq \tau_r^k \leq 1 \quad \forall p_i \in P, k = 1, \dots, K \quad (11)$$

Além disto, considera-se inicialmente um subconjunto do PLM, no qual  $R$  e  $R'$  são substituídos por conjuntos restritos  $\hat{R} \subset R$  e  $\hat{R}' \subset R'$  ( $|\hat{R}| \ll |R|$ ,  $|\hat{R}'| \ll |R'|$ ). A este problema da-se o nome de Problema Linear Master Restrito (PLMR)

## Definição das rotas iniciais

Os conjuntos  $\hat{R}$  e  $\hat{R}'$  são gerados através de um algoritmo guloso que define uma  $K$  partição viável em capacidade nos conjuntos  $S$  e  $C$  (solução básica viável para (1)-(7), (9)-(11))

# Geração de Colunas #1

- Resolve-se o PLMR e obtem-se os valores das variáveis duais associadas às restrições (2)-(7)

$$(2) \{\alpha^k \in \mathbb{R} : k = 1, \dots, K\}$$

$$(3) \{\beta^k \in \mathbb{R} : k = 1, \dots, K\}$$

$$(4) \{\theta_i \in \mathbb{R} : i \in S\}$$

$$(5) \{\mu_{i'} \in \mathbb{R} : i' \in C\}$$

$$(6) \{\pi_i^k \in \mathbb{R}_+ : k = 1, \dots, K, i \in S\}$$

$$(7) \{\chi_i^k \in \mathbb{R}_+ : k = 1, \dots, K, i \in P\}$$

- Novas variáveis  $r \in R \setminus \hat{R}$  ( $r' \in R' \setminus \hat{R}'$ ) podem ser adicionadas ao PLMR, caso violem as restrições

$$\alpha^k + \sum_{i \in S} a_r^i \theta_i + \sum_{i \in S} a_r^i \pi_i^k - \sum_{i \in S} a_r^i \chi_i^k \leq c_r \quad \forall r \in R \setminus \hat{R}, k = 1, \dots, K \quad (12)$$

$$\beta^k + \sum_{i' \in C} b_{r'}^{i'} \mu_{i'} - \sum_{i' \in C} b_{r'}^{i'} \pi_i^k + \sum_{i' \in C} b_{r'}^{i'} \chi_i^k \leq c_{r'} \quad \forall r' \in R' \setminus \hat{R}', k = 1, \dots, K \quad (13)$$

- Adiciona-se os custos  $-\theta_i - \pi_i^k + \chi_i^k$  aos vértices  $i \in S$  ( $-\mu_{i'} - \pi_i^k + \chi_i^k$  adicionados a  $i' \in C$ ) e o subproblema de geração de colunas consiste em resolver um caminho mínimo elementar com restrição de capacidade

# Geração de Colunas #1

- Para a solução dos subproblemas de geração de colunas foi implementado um algoritmo de Programação Dinâmica (PD) proposto por Feillet et al. (2004)
- O algoritmo utiliza *labels* para representar subcaminhos
  - Todos os *labels* iniciam no *Cross-Docking* e são estendidos para vértices adjacentes, cuja extensão seja viável
  - A extensão de um *label* a um adjacente origina outro label
  - Antes de um novo *label* ser incluído como candidato a caminho ótimo, ele passa por um teste de dominância

## Dominância de labels

O *label*  $l_1$  domina  $l_2$  caso ambos terminem no mesmo vértice,  $c_{l_1} < c_{l_2}$  e todas as possíveis extensões de  $l_2$  também sejam possíveis para  $l_1$

- O algoritmo investiga todos os *labels* não dominados que representem rotas viáveis e seleciona aquele de menor custo



# Branching #1

- A cada iteração,  $2K$  subproblemas são resolvidos e as rotas que violam (12) e (13) são adicionadas ao PLMR
- Quando não forem encontradas rotas que violem (12) e (13), a solução de PLM foi encontrada através do PLMR
- Caso a solução do PLMR seja inteira, ela também é ótima para (1)-(8) e o algoritmo para. Caso contrário:

Branching em variáveis de consolidação:  $\tau_i^k = \{0, 1\}$

# Branching #1

- A cada iteração,  $2K$  subproblemas são resolvidos e as rotas que violam (12) e (13) são adicionadas ao PLMR
- Quando não forem encontradas rotas que violem (12) e (13), a solução de PLM foi encontrada através do PLMR
- Caso a solução do PLMR seja inteira, ela também é ótima para (1)-(8) e o algoritmo para. Caso contrário:

Branching em variáveis de consolidação:  $\tau_i^k = \{0, 1\}$

nó	restrições associadas
<i>i</i>	$\tau_i^k = 0, \sum_{r \in R} a_r^i \lambda_r^k = 1, \sum_{r' \in R'} b_{r'}^{i'} \gamma_{r'}^k = 1$
<i>ii</i>	$\tau_i^k = 0, \sum_{r \in R} a_r^i \lambda_r^k = 0, \sum_{r' \in R'} b_{r'}^{i'} \gamma_{r'}^k = 0$
<i>iii</i>	$\tau_i^k = 1, \sum_{r \in R} a_r^i \lambda_r^k = 1, \sum_{r' \in R'} b_{r'}^{i'} \gamma_{r'}^k = 0$
<i>iv</i>	$\tau_i^k = 1, \sum_{r \in R} a_r^i \lambda_r^k = 0, \sum_{r' \in R'} b_{r'}^{i'} \gamma_{r'}^k = 1$

# Detalhes de Implementação #1

- O método de estabilização de pontos interiores foi implementado, a fim de acelerar a convergência do algoritmo em avaliar o PLM
- Foi implementada uma heurística para obter um limite primal inicial para o problema

## Heurística de Geração de Colunas

- Substitui os conjuntos  $R$  e  $R'$  do modelo (1)-(8) por  $\hat{R}$  e  $\hat{R}'$ , obtidos do PLMR ao avaliar o nó raiz
- Utiliza um solver para solucionar o Programa Inteiro Restrito associado
- O pacote de otimização CPLEX foi usado como LP e MIP solver

# Comentários

- Embora o modelo (1)-(8) trate de forma independente as rotas nos conjuntos  $S$  e  $C$ , permitindo reduzir a dimensão dos subproblemas, ele apresenta uma grande simetria
- Uma mesma rota  $r \in R$  ( $r' \in R'$ ) pode ser precificada para qualquer veículo, dificultando a convergência de BP#1
- Devido a esta simetria, as variáveis  $\tau_i^k$  assumem valor 0 em qualquer solução do PLM, independente do custo de troca  $c_i$
- Por este motivo, foi proposto um segundo modelo (e respectivo algoritmo) que não apresente esta simetria

## Formulação Matemática #2

- Neste modelo, as rotas do conjunto  $R$  consideram a visita dos veículos aos fornecedores e aos consumidores, incluindo o retorno ao *Cross-Docking* para a consolidação

### Parâmetros

- $c_r$ : soma dos custos dos arcos da rota  $r \in R$
- $a_{ir} \in \{0, 1\}$ : a rota  $r$  visita o vértice  $i$  ou não
- $b_{ir} = \max(0, a_{i'r} - a_{ir})$ : o consumidor  $i'$  é visitado pela rota  $r$ , mas seu respectivo fornecedor  $i$  não é

### Variáveis de decisão

- $\lambda_r \in \{0, 1\}$ : a rota  $r \in R$  é executada ou não por algum veículo na solução
- $\tau_i \in \{0, 1\}$ : a carga  $q_i$  é coletada e entregue pela mesma rota ou é coletada por uma rota e entregue por outra rota

# Formulação Matemática #2

$$\min \sum_{r \in R} c_r \lambda_r + \sum_{p_i \in P} c_i \tau_i \quad (14)$$

$$\sum_{r \in R} \lambda_r = K \quad (15)$$

$$\sum_{r \in R} a_{ir} \lambda_r = 1 \quad \forall i \in V \setminus \{0\} \quad (16)$$

$$\tau_i - \sum_{r \in R} b_{ir} \lambda_r \geq 0 \quad \forall p_i \in P \quad (17)$$

$$\lambda \in \mathbb{B}^{|R|}, \tau \in \mathbb{B}^n \quad (18)$$

## Geração de Colunas #2

- O PLM é obtido considerando-se o modelo (14)-(17) e

$$\lambda_r \geq 0 \quad \forall r \in R \quad (19)$$

$$0 \leq \tau_i \leq 1 \quad \forall p_i \in P \quad (20)$$

- No PLMR assume-se que um conjunto restrito de rotas  $\hat{R} \subset R$  ( $|\hat{R}| \ll |R|$ ) substitui o conjunto  $R$  no PLM (14)-(17), (19)-(20)
- O procedimento para geração de um conjunto inicial  $\hat{R}$  é similar ao descrito para o BP#1

## Geração de Colunas #2

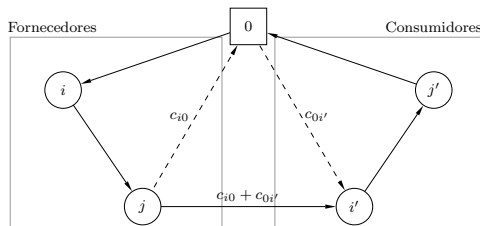
- Soluciona-se o PLMR e obtém-se os valores das variáveis duais associadas às restrições (15), (16) e (17)

$$(15) \{\alpha \in \mathbb{R}\}, \quad (16) \{\theta_i \in \mathbb{R} : i \in V \setminus \{0\}\}, \quad (17) \{\chi_i \in \mathbb{R}_+ : p_i \in P\}$$

- Procura-se por rotas  $r \in R \setminus \hat{R}$  que violem

$$\alpha + \sum_{i \in V \setminus \{0\}} a_{ir} \theta_i - \sum_{p_i \in P} b_i r \chi_i \leq c_r \quad (21)$$

- Considera-se os custos  $c'_{ij} = c_{ij} - \theta_j, \forall (i, j) \in A'$



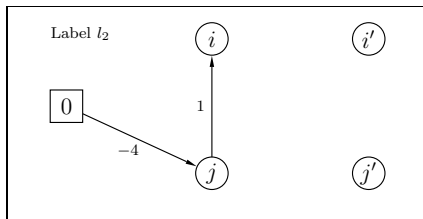
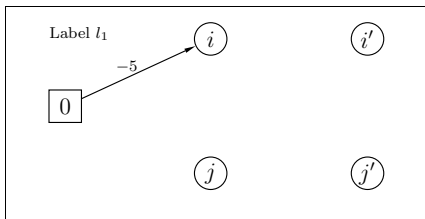


## Geração de Colunas #2

- O subproblema de geração de colunas é uma modificação do subproblema apresentado em BP#1
- O caminho mínimo deve considerar o custo de troca das cargas (caso ocorra), além dos custos nos arcos
- Novas regras de dominância devem ser implementadas no algoritmo de PD implementado inicialmente

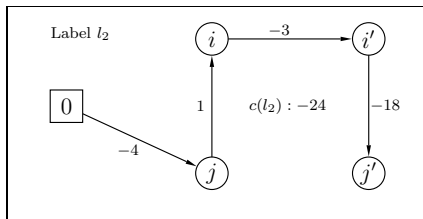
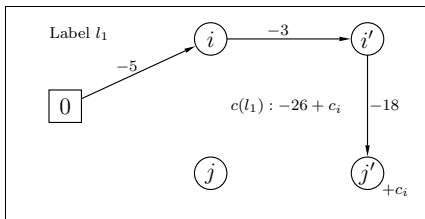
## Geração de Colunas #2

- O subproblema de geração de colunas é uma modificação do subproblema apresentado em BP#1
- O caminho mínimo deve considerar o custo de troca das cargas (caso ocorra), além dos custos nos arcos
- Novas regras de dominância devem ser implementadas no algoritmo de PD implementado inicialmente



# Geração de Colunas #2

- O subproblema de geração de colunas é uma modificação do subproblema apresentado em BP#1
- O caminho mínimo deve considerar o custo de troca das cargas (caso ocorra), além dos custos nos arcos
- Novas regras de dominância devem ser implementadas no algoritmo de PD implementado inicialmente



## Geração de Colunas #2

- As alterações realizadas nas regras de dominância do algoritmo de PD, fazem com que menos labels sejam descartados quando  $c_i > 0$
- Como consequência, o algoritmo tende a piorar seu desempenho na solução dos subproblemas
- Foram implementados algoritmos alternativos para solução dos subproblemas
  - Branch-and-cut (BC) adaptado de Jepsen et al. (2008)
  - Heurística GRASP

## Branching #2

- O processo de geração de colunas é executado enquanto existirem rotas que violem (21)
- Quando tais rotas não forem encontradas por PD ou BC, PLMR apresenta o limite de relaxação linear de PLM
- Caso todas as variáveis  $\lambda_r : r \in \hat{R}$  apresentem valores inteiros, PLMR apresenta a solução ótima para o VRPCD
- No contrário, executa-se *branching* nas variáveis de arco  $x_{ij}$ , obtidas do PLMR como

$$x_{ij} = \sum_{r \in \hat{R}} a_{ijr} \lambda_r$$

- Cria-se 2 nós na árvore de *branching* ao fixar  $x_{ij} = \{0, 1\}$

## Detalhes de Implementação #2

- Assim como em BP#1, a Heurística de Geração de Colunas foi implementada
  - Substitui  $R$  por  $\hat{R}$  no modelo (14)-(18) ao fim do nó raiz
- O algoritmo usa PD ou GRASP + BC para a solução dos subproblemas
- No segundo caso, BC é invocado apenas quando GRASP não é capaz de encontrar rotas de custo reduzido negativo
- Algoritmos codificados em C++, usando CPLEX como LP e MIP solver

# Resultados

## Instâncias

- Instâncias propostas por Wen et al.
  - Baseadas em informações reais, fornecidos por uma empresa de logística
  - Capacidade dos veículos é apertada
- Instâncias modificadas de Solomon
  - A modificação consiste em separar geograficamente fornecedores e consumidores
  - Veículos com capacidade para visitar um grande número de vértices
- Para ambos os grupos, foram extraídas instâncias com  $n = \{10, 15, 20, 25, 30\}$
- Os custos de troca foram empiricamente definidos como  $c_i = \{0, 20, 40\}$  para o primeiro grupo de instâncias e  $c_i = \{0, 10, 20\}$  para o segundo

# Resultados

- Compara-se o desempenho de BP#1 e BP#2 ao solucionar os dois conjuntos de instâncias
- Além dos algoritmos apresentados, foi proposto um modelo de fluxos para o VRPCD
- O modelo de fluxos, foi solucionado pelo usando o pacote MIP do CPLEX usando a opção de multi-threading com até 4 threads simultâneas
- Experimentos executados em um Intel Core 2 Quad 2.2GHz com 4 Gigas de memória RAM
- Limite de tempo de 4 horas imposto na execução do algoritmo



# Resultados

<i>n</i>	id	BP#1				BP#2				Fluxo
		<i>LP</i>	BLB	BUB	gap(%)	<i>LP</i>	BLB	BUB	gap(%)	gap(%)
10	a	<b>1659.9</b>	<b>1659.9</b>	<b>1659.9</b>	-	<b>1659.9</b>	<b>1659.9</b>	<b>1659.9</b>	-	-
	b	2064.3	<b>2130.1</b>	<b>2130.1</b>	-	2064.3	<b>2130.1</b>	<b>2130.1</b>	-	-
	c	1900.8	<b>1921.9</b>	<b>1921.9</b>	-	1900.8	<b>1921.9</b>	<b>1921.9</b>	-	-
	d	1748.0	<b>1776.9</b>	<b>1776.9</b>	-	1748.0	<b>1776.9</b>	<b>1776.9</b>	-	-
	e	1942.9	<b>1950.5</b>	<b>1950.5</b>	-	1942.9	<b>1950.5</b>	<b>1950.5</b>	-	-
15	a	<b>2594.0</b>	<b>2594.0</b>	<b>2594.0</b>	-	<b>2594.0</b>	<b>2594.0</b>	<b>2594.0</b>	-	2.6
	b	2892.7	2903.4	2929.6	0.89	2892.7	2892.8	2931.2	1.3	8.0
	c	2692.9	2703.7	2742.3	1.14	2692.9	2719.7	2742.3	0.83	9.2
	d	2622.4	2628.8	2679.7	1.9	2622.4	2625.0	2679.6	2.04	7.9
	e	2818.4	2821.4	2852.8	1.1	2818.4	2842.4	2852.8	0.36	6.2
30	a	4621.0	4621.0	4628.1	0.15	4621.0	4624.7	4628.1	0.07	N/A
	b	4321.0	4321.0	4389.9	1.57	4321.0	4323.5	4382.6	1.35	38.2
	c	4725.1	4725.1	4752.0	0.57	4725.1	4728.3	4752.3	0.51	39.6
	d	4178.6	4178.6	4209.3	0.73	4178.6	4178.6	4301.1	2.85	26.8
	e	4550.9	4550.9	4653.0	2.19	4550.9	4550.9	4660.9	2.36	36.3

**Tabela:** Instâncias adaptadas de Wen et. al,  $c_i = 0$ ,  $\forall p_i \in P$

# Resultados

<i>n</i>	id	BP#1				BP#2				Fluxo
		<i>LP</i>	BLB	BUB	gap(%)	<i>LP</i>	BLB	BUB	gap(%)	gap(%)
10	a	1659.9	<b>1809.8</b>	<b>1809.8</b>	-	<b>1809.8</b>	<b>1809.8</b>	<b>1809.8</b>	-	-
	b	2064.3	<b>2226.9</b>	<b>2226.9</b>	-	2208.5	<b>2226.9</b>	<b>2226.9</b>	-	-
	c	1900.8	<b>2017.5</b>	<b>2017.5</b>	-	<b>2017.5</b>	<b>2017.5</b>	<b>2017.5</b>	-	-
	d	1748.0	<b>1890.4</b>	<b>1890.4</b>	-	1888.0	<b>1890.4</b>	<b>1890.4</b>	-	-
	e	1942.9	<b>2124.0</b>	<b>2124.0</b>	-	2098.6	<b>2124.0</b>	<b>2124.0</b>	-	-
15	a	2594.0	2784.4	2802.2	0.64	<b>2802.2</b>	<b>2802.2</b>	<b>2802.2</b>	-	7.1
	b	2892.7	<b>3073.8</b>	<b>3073.8</b>	-	<b>3073.8</b>	<b>3073.8</b>	<b>3073.8</b>	-	4.7
	c	2692.9	2896.9	2938.5	1.41	2909.8	<b>2938.5</b>	<b>2938.5</b>	-	12.6
	d	2622.4	2768.0	2796.2	1.01	2786.3	<b>2796.2</b>	<b>2796.2</b>	-	9.6
	e	2818.4	3017.6	3027.9	0.34	3016.0	<b>3027.9</b>	<b>3027.9</b>	-	6.0
30	a	4621.0	4649.7	5068.1	8.2	4859.1	<b>4864.1</b>	<b>4864.1</b>	-	44.8
	b	4321.0	4382.5	5018.5	12.6	4577.5	4577.5	4678.9	2.17	45.9
	c	4725.1	4766.2	5150.7	7.4	5028.1	5028.2	5057.5	0.58	45.7
	d	4178.6	4217.6	4782.1	11.8	<u>4109.5</u>	4109.5	N/A	N/A	41.0
	e	4550.9	4582.4	5242.8	12.6	<u>4443.0</u>	4443.0	N/A	N/A	53.1

**Tabela:** Instâncias adaptadas de Wen et. al,  $c_i = 40$ ,  $\forall p_i \in P$

# Resultados

$n$	id	BP#1				BP#2				Fluxo
		LP	BLB	BUB	gap(%)	LP	BLB	BUB	gap(%)	gap(%)
10	c101	392.4	<b>420.1</b>	<b>420.1</b>	-	<b>420.1</b>	<b>420.1</b>	<b>420.1</b>	-	-
	r101	373.3	<b>373.8</b>	<b>373.8</b>	-	<b>373.8</b>	<b>373.8</b>	<b>373.8</b>	-	-
	rc101	385.1	<b>420.3</b>	<b>420.3</b>	-	<b>420.3</b>	<b>420.3</b>	<b>420.3</b>	-	-
15	c101	410.5	<b>417.7</b>	<b>417.7</b>	-	<b>417.7</b>	<b>417.7</b>	<b>417.7</b>	-	-
	r101	437.8	<b>455.0</b>	<b>455.0</b>	-	<b>455.0</b>	<b>455.0</b>	<b>455.0</b>	-	-
	rc101	435.2	<b>461.0</b>	<b>461.0</b>	-	<b>461.0</b>	<b>461.0</b>	<b>461.0</b>	-	-
20	c101	477.0	501.5	578.5	13.3	512.4	515.3	524.3	1.73	3.3
	r101	474.7	494.6	1087.2	54.5	506.6	<b>510.3</b>	<b>510.3</b>	-	-
	rc101	501.3	531.7	714.7	25.6	547.5	550.8	572.1	3.73	8.7
25	c101	545.8	554.5	766.9	27.7	616.7	617.0	618.3	0.2	22.5
	r101	553.3	559.6	800.2	30	602.0	602.0	619.7	2.85	9.7
	rc101	623.6	644.2	949.2	32.1	712.3	712.3	748.4	4.82	21.9

**Tabela:** Instâncias adaptadas de Solomon,  $c_i = 20$ ,  $\forall p_i \in P$

# Sumário

- 1 Introdução
- 2 VRPCD
- 3 PDPCD**
- 4 2EVRP
- 5 Cronograma

# Introdução

- No VRPCD, os veículos são obrigados a parar no *Cross-Docking* entre a coleta e entrega das cargas
- Os custos de entrega podem ser reduzidos ao realizar a consolidação das cargas, no entanto, existem casos onde o retorno ao *docking* pode fazer tais custos aumentarem
- A redução/diminuição dos custos de distribuição depende da posição geográfica dos fornecedores e consumidores

# Introdução

- Neste trabalho, propomos uma generalização do VRPCD, na qual o veículo pode evitar a consolidação das cargas no *Cross-Docking*
- Para isto, dois tipos de rotas são consideradas
  - **Rotas CD:** São as mesmas descritas para o VRPCD
  - **Rotas Diretas:** São rotas que coletam e entregam o mesmo conjunto de cargas, sem passar no CD para a consolidação. Estas rotas são as mesmas do Problema de Roteamento de Veículos com Coleta e Entrega
- A este novo problema, damos o nome de *Pickup and Delivery Problem with Cross-Docking - PDPCD*
- É proposta uma formulação matemática e um algoritmo BP para resolver o PDPCD

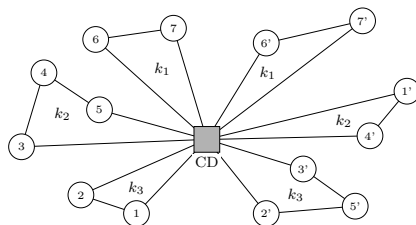
# Introdução

## PDPCD

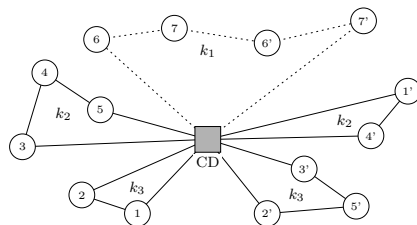
- $G = (\{S, C, 0\}, A_S \cup A_C \cup A_{SC})$ ,  
 $S = \{1, \dots, n\}$ ,  $C = \{1', \dots, n'\}$  e 0 são os fornecedores, consumidores e o *Cross-Docking*
- $A_S = \{(i, j) : i, j \in S \cup \{0\}\}$ ,  
 $A_C = \{(i, j) : i, j \in C \cup \{0\}\}$ ,  
 $A_{SC} = \{(i, j) : i \in S, j \in C\}$
- $n$  requisições dadas por  $P = \{p_i := (i, i', q_i) : i = 1, \dots, n\}$ :  
carga  $q_i > 0$  é coletada em  $i \in S$  e entregue em  $i' \in C$
- Uma frota homogênea de  $K$  veículos com capacidade  $Q$
- Custos  $c_{ij}$  nos arcos e  $c_i$  para trocar a carga  $q_i$  no CD

# Introdução

VRPCD



PDPCD





# Formulação Matemática

- Para se obter uma formulação para o PDPCD basta incluir o conjunto de rotas diretas viáveis  $R_d$  e uma variável binária  $\delta_r$ ,  $\forall r \in R_d$  no modelo (14)-(18)

$$\min \quad \sum_{r \in R} c_r \lambda_r + \sum_{r \in R_d} c_r \delta_r + \sum_{i=1}^n c_i \tau_i \quad (22)$$

$$\sum_{r \in R} \lambda_r + \sum_{r \in R_d} \delta_r = K \quad (23)$$

$$\sum_{r \in R} a_{ir} \lambda_r + \sum_{r \in R_d} a_{ir} \delta_r = 1 \quad i \in V \setminus \{0\} \quad (24)$$

$$\tau_i - \sum_{r \in R} b_{ir} \lambda_r \geq 0 \quad i = 1, \dots, n \quad (25)$$

$$\lambda \in \mathbb{B}^{|R|}, \delta \in \mathbb{B}^{|R_d|}, \tau \in \mathbb{B}^n. \quad (26)$$

# Geração de Colunas

- Discutiremos como o limite de relaxação linear do modelo (22)-(26) é avaliado

## PLM

É dado por (22)-(25) e

$$\lambda_r \geq 0 \quad \forall r \in R \quad (27)$$

$$\delta_r \geq 0 \quad \forall r \in R_d \quad (28)$$

$$0 \leq \tau_i \leq 1 \quad \forall p_i \in P \quad (29)$$

## PLMR

Obtido ao substituir  $R$  e  $R_d$  no PLM por conjuntos restritos de rotas CD e diretas  $\hat{R} \subset R$  e  $\hat{R}_d \subset R_d$  ( $|\hat{R}| \ll |R|$ ,  $|\hat{R}_d| \ll |R_d|$ )

# Geração de Colunas

- A condição de otimalidade do PLM é escrita usando-se as variáveis duais associadas às restrições (23), (24) e (25) ao solucionar o PLMR

$$(23) \{\alpha \in \mathbb{R}\} \quad (24) \{\theta_i \in \mathbb{R} : i \in V \setminus \{0\}\} \quad (25) \{\chi_i \in \mathbb{R}_+ : p_i \in P\}$$

## Condição de otimalidade do PLM

$$\alpha + \sum_{i \in V \setminus \{0\}} a_{ir} \theta_i - \sum_{i=1}^n b_{ir} \chi_i \leq c^r \quad \forall r \in R \setminus \hat{R} \quad (30)$$

$$\alpha + \sum_{i \in V \setminus \{0\}} a_{ir} \theta_i \leq c^r \quad \forall r \in R_d \setminus \hat{R}_d \quad (31)$$

# Geração de Colunas

- Dois subproblemas devem ser resolvidos para a precificação de rotas no PLMR
- O primeiro subproblema é o mesmo descrito para o algoritmo BP#2 na solução do VRPCD
- O segundo é uma modificação do subproblema de BP#1, o caminho mínimo elementar com restrição de capacidade
  - Um consumidor pode ser visitado apenas se o respectivo fornecedor também for
  - O custo dos arcos que conectam os fornecedores é dado por  $c_{ij} - \theta_j - \theta_{j'}$ . Para os consumidores o custo nos arcos é  $c_{ij}$
  - Foi implementado um algoritmo de PD que apresenta um bom desempenho, devido às particularidades do problema

# Detalhes de implementação

- Durante a geração de colunas, as rotas diretas são precificadas antes das rotas CD
- Após avaliar o PLM, caso a solução tenha variáveis não-inteiras, executa-se *branching* em arcos

$$\sum_{r \in \hat{R}} a_{ijr} \lambda_r + \sum_{r \in \hat{R}_d} a_{ijr} \delta_r = \{0, 1\}$$

- Como nos algoritmos anteriores, implementa-se também uma Heurística de Geração de Colunas para ser executada no nó raiz
- CPLEX usado como LP e MIP solver

# Resultados

- Mesmo cenário de experimentos usado para o VRPCD
  - Programas codificados em C++ usando CPLEX como LP e MIP solver
  - Máquina de execução dos experimentos Intel Core 2 Quad 2.2GHz e 4GB de RAM
  - Apenas as instâncias adaptadas de Wen et al. foram consideradas
- Cada instância foi solucionada usando o BP para o PDPCD e o BP#2 para o VRPCD
- Custos de troca  $c_i = 20$ ,  $\forall p_i \in P$

instance		BP for PDPCD					VRPCD		
		BLB	BUB	time (s)	# routes		BLB	BUB	time (s)
<i>n</i>	id				PD	CD			
10	<i>a</i>	1687.5	1687.5	0.56	1	3	1709.7	1709.7	0.42
	<i>b</i>	2083.2	2083.2	186	2	2	2141.7	2166.9	tl
	<i>c</i>	1902.5	1902.5	12	1	3	1951.9	1951.9	2
	<i>d</i>	1775.9	1775.9	0.38	1	3	1813.5	1813.5	86
	<i>e</i>	1987.1	1987.1	199	1	3	2000.5	2000.5	17
15	<i>a</i>	2622.2	2622.2	262	2	4	2654.0	2654.0	0.88
	<i>b</i>	2925.1	2925.1	1	2	4	2981.0	2983.8	tl
	<i>c</i>	2722.6	2722.6	8991	3	3	2802.2	2814.9	tl
	<i>d</i>	2597.4	2597.4	2	4	2	2707.2	2729.3	tl
	<i>e</i>	2892.0	2892.0	912	1	5	2916.8	2916.8	1126
20	<i>a</i>	3213.5	3256.4	tl	7	0	3273.1	3327.1	tl
	<i>b</i>	3288.5	3288.5	3592	4	3	3411.5	3483.6	tl
	<i>c</i>	2999.2	2999.2	1652	5	2	3147.8	3161.9	tl
	<i>d</i>	3333.8	3343.7	tl	3	4	3401.1	3503.6	tl
	<i>e</i>	3044.9	3044.9	117	4	3	3105.2	3120.8	tl
30	<i>a</i>	4618.2	4621.4	tl	6	4	4729.8	4731.1	tl
	<i>b</i>	4286.5	4335.2	tl	8	2	4449.6	4540.3	tl
	<i>c</i>	4706.6	4706.6	13728	10	0	4873.8	4959.8	tl
	<i>d</i>	4191.7	4240.5	tl	8	2	4278.4	4396.7	tl
	<i>e</i>	4569.5	4645.3	tl	10	0	4668.9	4846.6	tl

# Sumário

- 1 Introdução
- 2 VRPCD
- 3 PDPCD
- 4 2EVRP**
- 5 Cronograma



# Introdução

- Roteamento multi-níveis é uma estratégia cada vez mais usada em sistemas de logística
- As mercadorias partem da origem e passam por armazéns intermediários antes de chegar ao destino As mercadorias devem passar por depósitos intermediários
- Por que utilizar roteamento multi-níveis
  - Restrições quanto ao fluxo de veículos pesados
  - Redução dos custos de transporte
  - Restrições ambientais
- Os armazéns intermediários são denominados satélites

# Introdução

## Satélites

- Recebe mercadorias de um conjunto de veículos de entrada
- As mercadorias são consolidadas e encaminhadas a um conjunto de veículos de saída
- A manipulação das mercadorias pode ter um custo associado
- Restrições comuns na operação dos satélites
  - Inexistência de estoque de mercadorias
  - Limites quanto ao número de veículos de entrada e saída

# Introdução

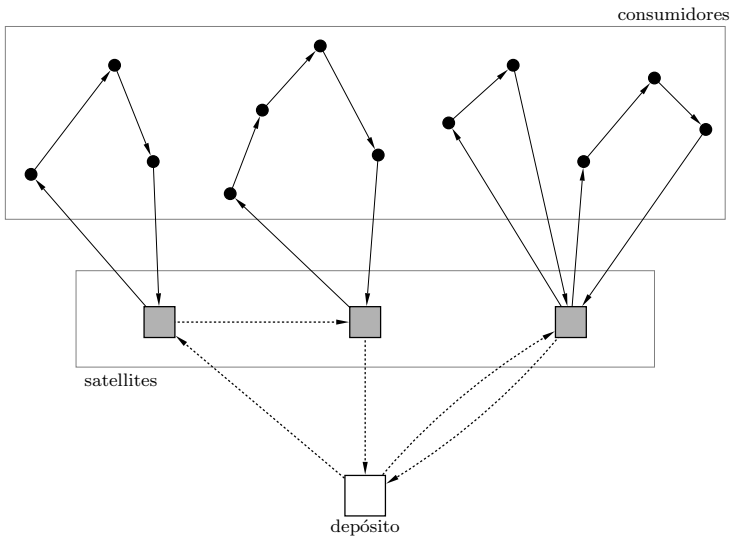
- O roteamento em 2 níveis é o mais comum entre os esquemas multi-níveis
- Um conjunto de mercadorias parte do depósito para serem entregues aos consumidores e usam exatamente um satélite como ponto intermediário para a consolidação
- Na consolidação as mercadorias devem passar dos veículos do primeiro nível para veículos do segundo
- A este problema denomina-se Problema de Roteamento de Veículos com Dois Elos (Two-Echelon Vehicle Routing Problem - 2EVRP)

# Introdução

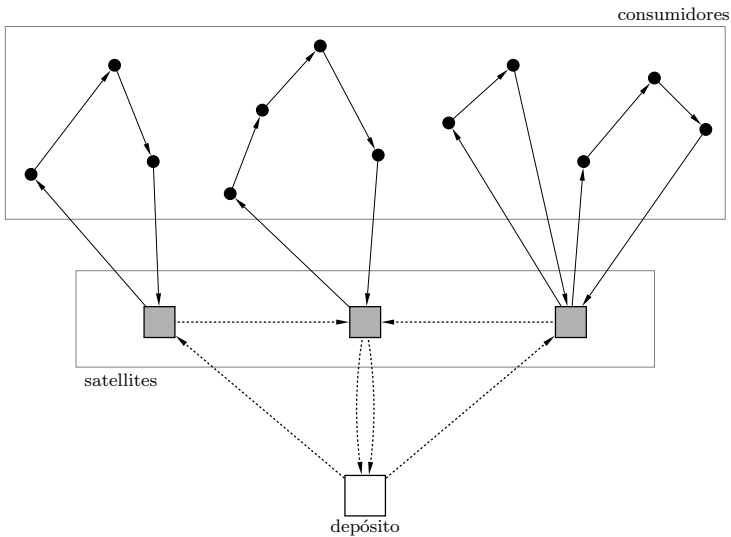
## 2EVRP

- Definido no grafo direcionado  $G = (\{\{0\}, S, C\}, \{A_1, A_2\})$
- $S = \{s_1, \dots, s_k\}$  é o conjunto dos satélites,  $C = \{c_1, \dots, c_n\}$  denota os consumidores e 0 o depósito
- $A_1 = \{(i, j) : i, j \in S \cup \{0\}\}$  é o conjunto de arcos do nível 1  
 $A_2 = \{(i, j) : i, j \in C\} \cup \{(i, j) : i \in C, j \in S \text{ ou } i \in S, j \in C\}$  denota o conjunto de arcos do nível 2
- Cada consumidor apresenta uma demanda  $q_i > 0$ , enquanto os satélites devem ter balanço de carga 0
- $K_1$  e  $K_2$  veículos de capacidade  $Q_1$  e  $Q_2$  são disponíveis respectivamente para o nível 1 e 2
- No máximo  $m_s$  veículos podem sair do satélite  $s$
- Custos  $c_{ij} : (i, j) \in A_1 \cup A_2$  nos arcos e  $L_s$  para consolidar uma unidade de carga no satélite  $s$

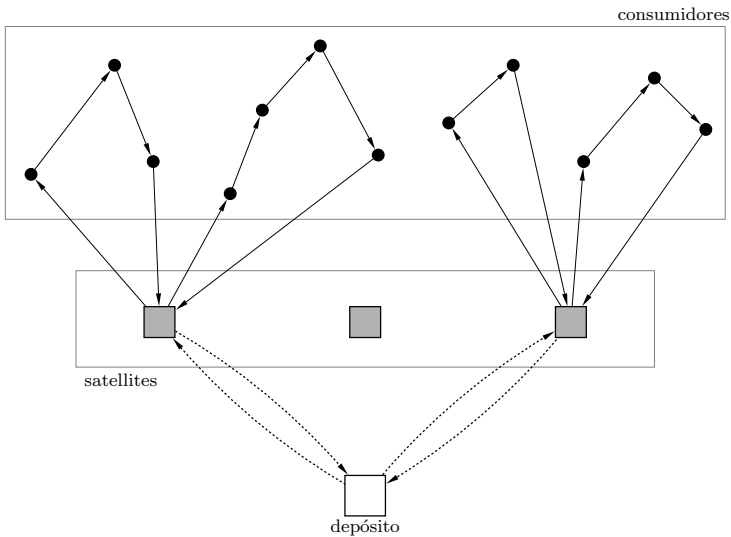
# Introdução



# Introdução



# Introdução



# Introdução

## Trabalhos Relacionados

- O 2EVRP foi proposto por Perboli et al. (2008, 2011).
    - Os autores introduziram o problema, apresentaram uma formulação matemática, um algoritmo BC e duas heurísticas
  - Perboli e Vigo (2010) apresentaram novas famílias de desigualdades para a formulação de Perboli et al. (2008)
  - Crainic et al. (2010) apresentaram um conjunto de heurísticas *multi-start* para o problema
- 
- Neste trabalho propomos uma formulação matemática e um algoritmo BP para solucionar o 2EVRP



# Formulação Matemática

- Na formulação, dois conjuntos de rotas são considerados
  - $R_1$  contém as rotas de nível 1
  - $R_2^s$  contém as rotas de nível 2 que partem do satélite  $s$
- As rotas em  $R_2^s$  são elementares e viáveis em capacidade e as rotas em  $R_1$  desconsideram a capacidade

## Parâmetros

- $c_r$ : custo da rota de nível 1 ou 2
- $a_{ir} \in \{0, 1\}$ : a rota  $r$  visita o vértice  $i$  ou não

## Variáveis de decisão

- $\lambda_r^k \in \{0, 1\}$ : o veículo  $k$  executa a rota  $r \in R_1$  ou não
- $\gamma_r^s \in \{0, 1\}$ : a rota  $r \in R_2^s$  é executada ou não na solução
- $\delta^{sk} \in \mathbb{N}$ : quantidade de carga enviada do depósito ao satélite  $s$  usando o veículo  $k$

# Formulação Matemática

$$\min \sum_{k=1}^{K_1} \sum_{r \in R_1} c_r \lambda_r^k + \sum_{s \in S} \sum_{r \in R_2^s} c_r \gamma_r^s + \sum_{k=1}^{K_1} \sum_{s \in S} L_s \delta^{sk} \quad (32)$$

$$\sum_{k=1}^{K_1} \sum_{r \in R_1} \lambda_r^k \leq K_1 \quad (33)$$

$$\sum_{s \in S} \sum_{r \in R_2^s} \gamma_r^s \leq K_2 \quad (34)$$

$$\sum_{r \in R_2^s} \gamma_r^s \leq m_s \quad \forall s \in S \quad (35)$$

$$\sum_{s \in S} \sum_{r \in R_2^s} a_{ir} \gamma_r^s = 1 \quad \forall i \in C \quad (36)$$

$$\sum_{r \in R_2^s} \gamma_r^s \sum_{i \in C} a_{ir} d_i = \sum_{k=1}^{K_1} \delta^{sk} \quad \forall s \in S \quad (37)$$

$$\delta^{sk} \leq Q_1 \sum_{r \in R_1} a_{sr} \lambda_r^k \quad \forall s \in S, k = \{1, \dots, K_1\} \quad (38)$$

$$\sum_{s \in S} \delta^{sk} \leq Q_1 \quad \forall k = \{1, \dots, K_1\} \quad (39)$$

$$\lambda \in \mathbb{B}^{K_1 |R_1|}, \gamma \in \mathbb{B}^{|S| |R_2|}, \delta \in \mathbb{N}^{|S| K_1} \quad (40)$$

# Geração de Colunas

## Fortalecimento do modelo (32)-(40)

$$\sum_{k=1}^{K_1} \sum_{r \in R_1} \lambda_r^k \geq \left\lceil \frac{\sum_{i \in C} d_i}{Q_1} \right\rceil \quad (41)$$

- O PLM é obtido ao considerar a formulação (32)-(39), (41) e

$$\lambda_r^k \geq 0 \quad \forall r \in R_1, k = \{1, \dots, K_1\} \quad (42)$$

$$\gamma_r^s \geq 0 \quad \forall r \in R_2^s, s = \{s_1, \dots, s_k\} \quad (43)$$

$$\delta^{sk} \geq 0 \quad s = \{s_1, \dots, s_k\}, k = \{1, \dots, K_1\} \quad (44)$$

- No PLMR considera-se o modelo (32)-(39), (41)-(44) e substitui os conjuntos  $R_2^s$  por conjuntos restritos  $\hat{R}_2^s$
- As rotas do conjunto  $R_1$  serão enumeradas explicitamente

# Geração de Colunas

- Após solucionar o PLMR e obter os valores das variáveis duais associadas às restrições (34)-(37)

$$(34) \beta \in \mathbb{R}_-$$

$$(35) \{\tau_s \in \mathbb{R}_- : s \in S\},$$

$$(36) \{\theta_i \in \mathbb{R} : i \in C\}$$

$$(37) \{\mu_s \in \mathbb{R} : s \in S\}$$

O

subproblema de geração de colunas consiste em encontrar rotas que violem as restrições duais

$$\beta + \tau_s + \sum_{i \in C} a_{ir} \theta_i + \sum_{i \in C} a_{ir} d_i \mu_s \leq c_r \quad \forall s \in S, \forall r \in R_2^s \setminus \hat{R}_2^s \quad (45)$$

- Adiciona-se custos duais  $-\theta_i - d_i \mu_s$  aos vértices  $i \in C$  e o subproblema é o caminho mínimo elementar com restrição de capacidade
  - Solucionado usando o algoritmo BC sem custos de troca

# Geração de Colunas

## Solução alternativa para o subproblema

- Substituir os conjuntos  $R_2^s$  por  $\mathcal{R}_2^s$  no modelo (32)-(40) que incluem as rotas elementares e não-elementares (exceto ciclos  $i - j - i$ )
- Devido à desigualdade triangular nos arcos, apenas rotas elementares estarão presentes na solução ótima
- Embora  $|\mathcal{R}_2^s| \gg |R_2^s|$ , existe algoritmo pseudo-polinomial para precificar rotas em  $\mathcal{R}_2^s \setminus \hat{\mathcal{R}}_2^s$
- Os limites duais usando os conjuntos  $\mathcal{R}_2^s$  tendem a ser mais fracos, mas a eficiência do algoritmo para resolver os subproblemas pode compensar

# Geração de Colunas

## Solução do subproblema em $\mathcal{R}_2^s$

- As rotas em  $\mathcal{R}_2^s$  são conhecidas como *q-routes*
- Um algoritmo de PD para encontrar *q-routes* foi proposto por Christofides et al (1981)
- O algoritmo cria um estado para armazenar o caminho mínimo entre  $s$  e  $j \in C \cup \{s\}$  considerando a capacidade máxima  $q = \{1, \dots, Q_2\}$

- A relação de recorrência é dada por

$$p[j, q] = \begin{cases} 0 & j = s \\ \infty & q - d_j < 0 \\ \min_{i \in C \cup \{s\}} \{p[i, q - d_j] + c_{ij}, p[j, q - d_j]\} & q - d_j \geq 0 \end{cases}$$

- Para a implementação do algoritmo, utiliza-se uma matriz  $|C| + 1 \times Q_2$ 
  - Linhas representam vértices  $j$  e colunas a capacidade  $q$

# Branching

- Independente do conjunto de rotas usado, o processo de geração de colunas para quando não existirem rotas que violem (45) para todos os satélites  $s \in S$
- Branching em arcos de rotas que partem do satélite  $s$

$$\sum_{r \in \hat{R}_2^s} a_{ijr} \gamma_r^s = \{0, 1\}$$

- Executa-se a geração de colunas em cada nó, até que não existam mais nós a serem processados
- Como nos outros algoritmos, usa-se a Heurística de Geração de Colunas no nó raiz
- CPLEX usado como MIP e LP solver

# Resultados

- Mesmo conjunto de instâncias usado em Perboli et al (2008, 2011), Perboli e Vigo (2010) e Crainic et al (2010)
- O resultado dos algoritmos é comparado a uma compilação dos melhores limites duais e primais da literatura
- O algoritmo que precifica rotas elementares em  $R_2^S$  é rotulado como  $BP^E$ , enquanto  $BP^N$  identifica o algoritmo que precifica rotas em  $\mathcal{R}_2^S$
- Limite de tempo de 10000 segundos é imposto na execução dos algoritmos



# Resultados

instance	BP <sup>E</sup>		BP <sup>N</sup>		BP gap(%)	BC+H		
	BLB	BUB	BLB	BUB		BLB	BUB	gap(%)
n51-s2-17	562.9	597.4	561.9	-	5.8	556.5	597.5	6.8
n51-s4-46	527.7	530.7	530.7	530.7	0.0	529.3	530.7	0.2
n51-s6-12	536.8	557.2	537.2	-	3.6	541.1	554.8	2.4
n51-s11-19	574.8	591.5	576.7	618.4	2.5	558.2	581.6	4.0
n51-s27-47	521.0	538.2	524.5	573.5	2.5	535.0	538.2	0.5
n51-s32-37	534.3	552.2	539.5	-	2.3	552.2	552.2	0.0
n51-s2-4-17-46	508.2	530.7	530.7	530.7	0.0	515.7	541.0	4.6
n51-s6-12-32-37	509.3	531.9	520.6	-	2.1	516.0	538.8	4.2
n51-s11-19-27-47	510.0	527.6	523.0	-	0.9	511.0	531.2	3.8
n50-s5-37	1379.3	1542.3	1463.8	-	5.1	1434.5	1548.0	7.3
n50-s5-39	1364.4	1533.2	1477.6	-	3.6	1423.4	1525.2	6.6
n50-s5-41	1521.6	1698.4	1609.2	1808.7	5.3	1580.8	1703.0	7.2
n50-s5-43	1321.5	1456.9	1308.9	-	9.3	1341.0	1453.1	7.7
n50-s5-45	1324.0	1495.8	1348.9	-	9.8	1331.5	1484.6	10.3
n50-s5-47	1459.8	1608.2	1491.7	-	7.2	1487.0	1598.8	6.9
n50-s5-49	1304.9	1496.1	1375.5	-	8.1	1370.8	1479.1	7.3
n50-s5-51	1253.3	1429.1	1360.9	1617.7	4.8	1289.4	1436.3	10.2
n50-s5-53	1418.7	1614.2	1501.7	-	7.0	1483.1	1552.7	4.4

# Resultados

instance	BP <sup>E</sup>		BP <sup>N</sup>		BP gap(%)	BC+H		
	BLB	BUB	BLB	BUB		BLB	BUB	gap(%)
n33-s16-22	641.7	674.7	639.7	712.0	4.9	634.2	672.1	5.6
n33-s16-24	650.2	669.1	643.6	-	2.8	622.8	668.8	6.8
n33-s19-26	662.0	680.3	653.0	-	2.7	648.1	680.3	4.9
n33-s22-26	661.5	680.3	658.2	699.4	2.8	650.3	680.7	4.6
n33-s24-28	663.0	670.4	655.7	701.3	1.1	633.6	672.3	5.7
n33-s25-28	637.8	651.3	624.6	-	2.1	616.5	651.2	5.4
n51-s12-18	700.5	733.7	698.1	-	4.5	661.3	705.5	6.6
n51-s12-41	691.9	731.9	686.1	-	5.5	635.6	716.5	11.2
n51-s12-43	677.6	717.5	677.8	874.2	5.5	688.0	710.3	3.2
n51-s39-41	697.2	739.3	705.2	-	4.6	679.8	729.9	6.8
n51-s40-41	695.1	732.1	695.0	-	5.1	669.7	732.4	8.6
n51-s40-43	677.8	717.9	680.1	867.4	5.3	707.8	757.3	6.5

# Resultados

- $BP^N$  avalia soluções ótimas para instâncias de pequeno porte mais rápido que BC
  - Para solucionar um conjunto de 66 instâncias com  $|C| = 12$  e  $|S| = 2$ ,  $BP^N$  gasta 302 segundos contra 3788 de BC
- $BP^N$  obtém certificado de otimalidade para duas novas instâncias
- Entre as 27 instâncias não resolvidas na otimalidade por nenhum dos algoritmos
  - em 22 os melhores limites duais são obtidos por BP
  - em 12 os melhores limites primais são obtidos por BP
  - em 20 os gaps de dualidade obtidos por BP são menores que os obtidos pelos outros algoritmos da literatura

# Sumário

- 1 Introdução
- 2 VRPCD
- 3 PDPCD
- 4 2EVRP
- 5 Cronograma**

# Cronograma

- 1 Investigação de planos de corte para o 2EVRP
- 2 Considerar custos para a utilização dos satélites
- 3 Introdução de janelas de tempo no VRPCD e PDPCD
- 4 Escrita da tese
- 5 Defesa da tese

Tarefa	04/2012	05/2012	06/2012	07/2012	08/2012	09/2012
1	■	■	■			
2			■			
3				■	■	
4		■	■	■	■	
5						■

# À disposição da banca