

Otimização de Processos de Negócios em Cadeia de Suprimentos

Edcarllos Santos

Email:

esantos@ic.uff.br

Universidade Federal Fluminense
Instituto de Computação
Niterói - RJ

Outubro/2016



Conteúdo

- 1 Introdução
- 2 Logística
- 3 PERVMP
- 4 Algoritmo HILS-RVND
- 5 Resultados Computacionais
- 6 Conclusão e trabalhos futuros

Conteúdo

- 1 Introdução
- 2 Logística
- 3 PERVMP
- 4 Algoritmo HILS-RVND
- 5 Resultados Computacionais
- 6 Conclusão e trabalhos futuros

Agropecuária

É responsável por...

- ▶ produzir alimentos para a população
- ▶ fornecer matérias-primas à indústria
- ▶ fornecer divisas através das exportações



Agropecuária no Brasil

- ▶ responsável por cerca de **25%** do PIB brasileiro
- ▶ o país possui diversas vantagens:
 - ▶ clima favorável
 - ▶ relevo pouco acidentado
 - ▶ solo fértil
 - ▶ grande mercado consumidor interno e externo

Agropecuária no Brasil



Alguns dados sobre a agropecuária no Brasil...

- ▶ 2º maior produtor de carne bovina
- ▶ 4º maior produtor de leite
- ▶ Maior exportador de frango
- ▶ 2º maior exportador de soja
- ▶ 4º país em exportação de suínos

Previsões ainda afirmam que...

- ▶ Em 5 anos, pode se tornar o maior produtor de carne bovina
- ▶ Em 10 anos, pode se tornar o maior produtor de soja
- ▶ Brasil será **maior exportador** agrícola mundial em 2024

Agropecuária no Tocantins

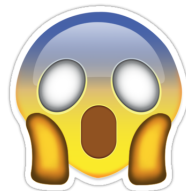
- ▶ principal atividade econômica privada de 72,6% dos municípios
 - ▶ Pedro Afonso
 - ▶ Lagoa da Confusão
 - ▶ Formoso do Araguaia
- ▶ maior produtor de grãos da região Norte do Brasil
 - ▶ arroz (3º no Brasil)
 - ▶ soja
 - ▶ melancia
 - ▶ milho
 - ▶ feijão
- ▶ tem o menor índice de desmatamento na região Norte



Agropecuária no mundo

Estudos e previsões afirmam...

- ▶ em 2050 será necessário 60% a mais de comida
- ▶ o consumo de alimentos deve dobrar nas próximas quatro décadas no mundo



O problema...

- ▶ soluções existentes não são ideais para o cenário futuro

A solução...

- ▶ usar tecnologias para aumentar a produtividade para a produção de alimentos

Conteúdo

1 Introdução

2 Logística

3 PERVMP

4 Algoritmo HILS-RVND

5 Resultados Computacionais

6 Conclusão e trabalhos futuros

Cadeia de Suprimentos

O que é?

- ▶ conjunto de atividades funcionais ^a que se repetem inúmeras vezes ao longo do canal pelo qual matérias-primas vão sendo convertidas em produtos acabados

^ae.g. transporte, controle de estoque e etc

Quem participa?

- ▶ fornecedores de matérias primas
- ▶ fabricantes
- ▶ atacadista e distribuidores
- ▶ varejistas
- ▶ clientes



Cadeia de suprimentos



Gestão da cadeia de suprimentos

O que é logística?

- ▶ administração das atividades funcionais (*aka* processos de negócios) que facilitam a movimentação e coordenação de bens desde sua criação ao armazenamento



Processos de negócios

- ▶ distribuição de bens
- ▶ armazenamento de produtos
- ▶ contabilidade gerencial
- ▶ previsão de vendas

Logística

Que fatores afetam a gerência dos processos de negócios?

- ▶ Cultura e regras governamentais
 - ▶ Mudanças de impostos e incentivos fiscais
- ▶ Globalização
 - ▶ Acordos regionais como Mercosul, Nafta e outros
- ▶ Mercado e demanda
 - ▶ Necessidade x Disponibilidade x Qualidade x Preço
- ▶ Meio ambiente
 - ▶ Valorização e preservação
- ▶ Competição intensa
 - ▶ Serviços diferenciados
- ▶ Informação e comunicação
 - ▶ Marketing social

A dinamicidade do processo logístico

- ▶ Objetivos conflitantes

Produção: alta qualidade x alta produtividade x baixo custo de produção

Estoque: baixo nível de estoque x capacidade rápida de reposição

Cliente: curtos prazos de entrega x preços baixos x grande quantidade em estoque

- ▶ Complexa rede de instalações

- ▶ Variações temporais do sistema (incerteza)

Vantagens de uso da logística

Redução de custos

- ▶ **Objetivo:** maximização de lucros
- ▶ Reduzir custos variáveis de transporte e armazenagem
 - ▶ escolha do local de armazenagem



Redução de capital

- ▶ **Objetivo:** maximização do retorno sobre os ativos logísticos
- ▶ Reduzir nível de investimento no sistema logístico
 - ▶ terceirização de serviços logísticos

Melhoria dos serviços

- ▶ **Objetivo:** estabelecer o nível apropriado de serviços aos clientes
- ▶ Aumentar lucros
 - ▶ redução do tempo de entrega



Logística

No Brasil...

- ▶ custos logísticos correspondem a 10,6% do PIB
- ▶ para empresas, gastos com logística representam 8,5% da receita líquida

Transporte de mercadorias

- ▶ representa entre metade e 2/3 dos custos logísticos totais
 - ▶ seleção de modal
 - ▶ consolidação de fretes
 - ▶ roteirização
 - ▶ programação de veículos



Estoque de produtos

- ▶ funcionam como “pulmão” entre oferta e demanda
 - ▶ estratégias de estocagem de matérias-primas e produtos
 - ▶ previsão de vendas a curto prazo
 - ▶ número, tamanho e localização das instalações

Conteúdo

1 Introdução

2 Logística

3 **PERVMP**

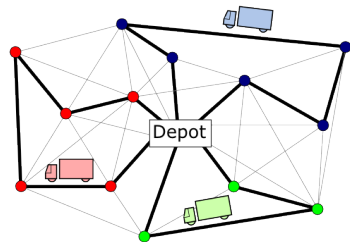
4 Algoritmo HILS-RVND

5 Resultados Computacionais

6 Conclusão e trabalhos futuros

Problema de Estoque e Roteamento de Veículos (PERV)

- ▶ Extensão do Problema de Roteamento de Veículos
- ▶ \mathcal{NP} -difícil



Decisões a estipular...

- 1 definir os dias de atendimentos para cada cliente
- 2 atribuir a quantidade de produtos a serem entregues em cada atendimento
- 3 encontrar as melhores rotas para atender os clientes

Objetivo

Minimizar os custos operacionais de distribuição e estocagem de produtos ao longo de um horizonte de planejamento

PERV com Múltiplos Veículos (PERVMP)

ATRIBUTOS		OPÇÕES	
Topologia de rede	1-1	1-n	n-n
Horizonte de planejamento	Finito	Infinito	-
Tipo de produtos	Único	Múltiplos	-
Tipo de demanda	Determinístico	Estocástico	-
Composição da frota	Homogênea	Heterogênea	Irrestrito
Tamanho da frota	Fixa	Variável de decisão	Irrestrita
Gestão do estoque	<i>Backlogging</i>	<i>Lost sales</i>	<i>No negative</i>

PERV com Múltiplos Veículos (PERVMP)

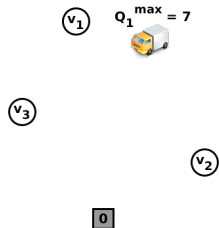
- ▶ Um depósito para suprir todos os clientes
- ▶ Demandas determinísticas para um único tipo de produto
- ▶ Frota de veículos homogênea e limitada
- ▶ Estoques independentes para fornecedor e clientes

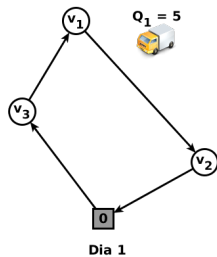
Objetivo

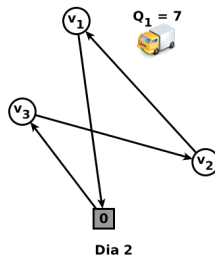
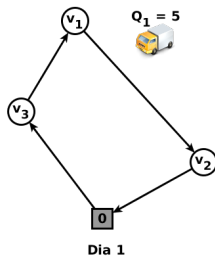
- ▶ Minimizar custos de distribuição e estocagem de produtos

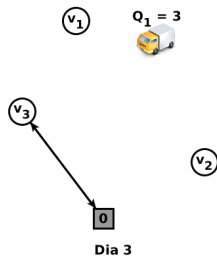
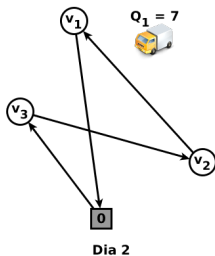
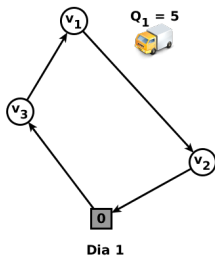
Restrições

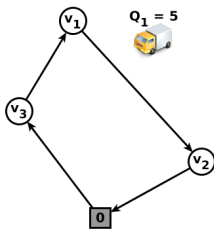
- ▶ Rotas iniciam e terminam no fornecedor sem violar a capacidade do veículo
- ▶ Falta de estoque não é permitido
- ▶ Capacidade de estoque deve ser respeitada



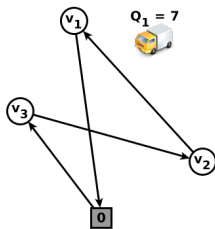




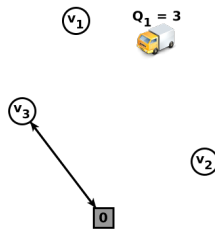




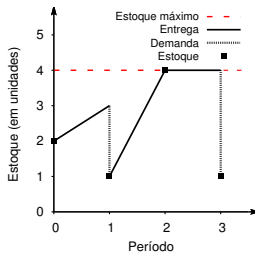
Dia 1

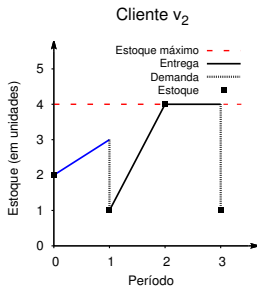
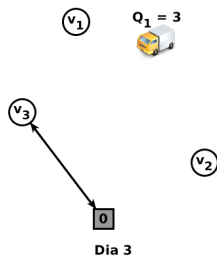
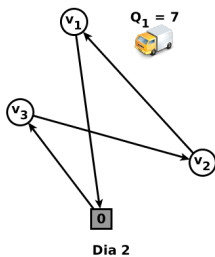
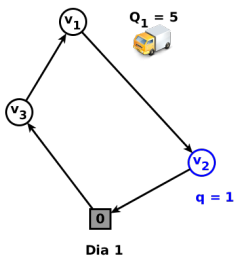


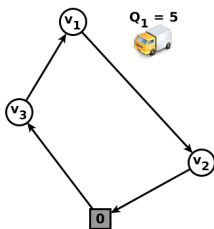
Dia 2



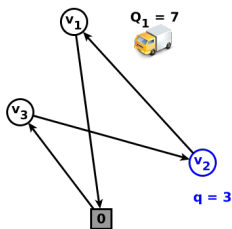
Dia 3

Cliente v_2 

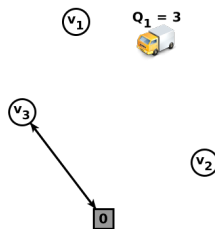




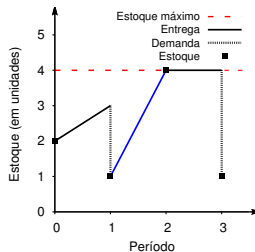
Dia 1

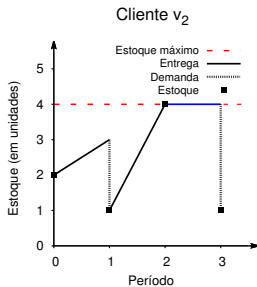
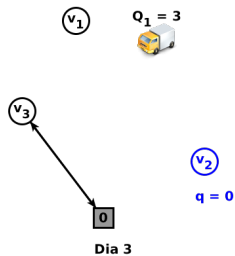
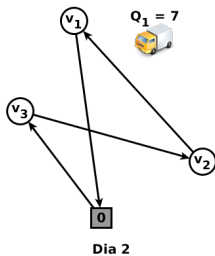
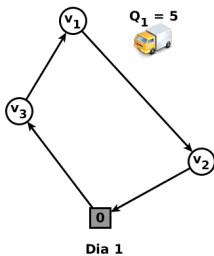


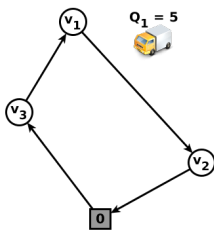
Dia 2



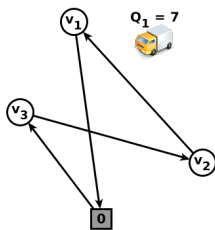
Dia 3

Cliente v_2 

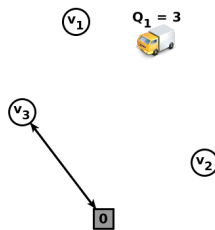




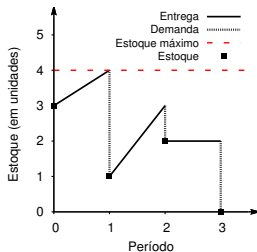
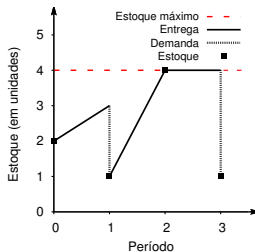
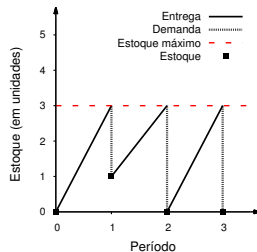
Dia 1



Dia 2



Dia 3

Cliente v_1 Cliente v_2 Cliente v_3 

Conteúdo

- 1 Introdução
- 2 Logística
- 3 PERVMP
- 4 Algoritmo HILS-RVND**
- 5 Resultados Computacionais
- 6 Conclusão e trabalhos futuros

Metaheurísticas

Heurísticas

- ▶ são regras gerais de influência utilizadas pelo decisor para simplificar seus julgamentos em tarefas decisórias de incerteza

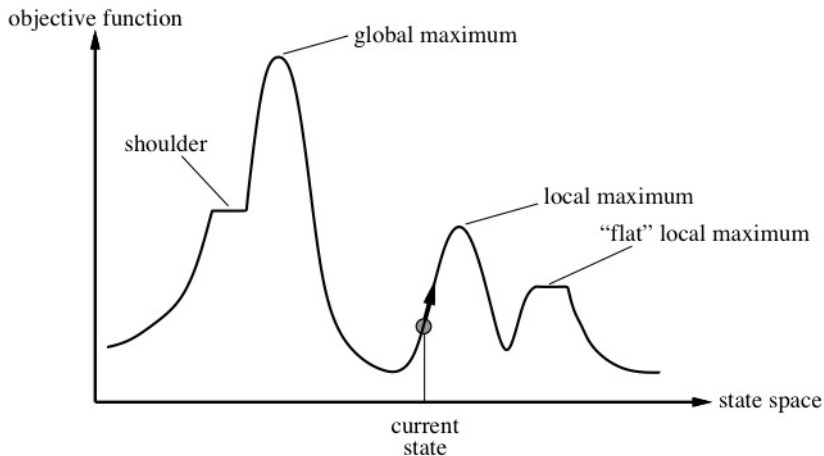
Metaheurísticas

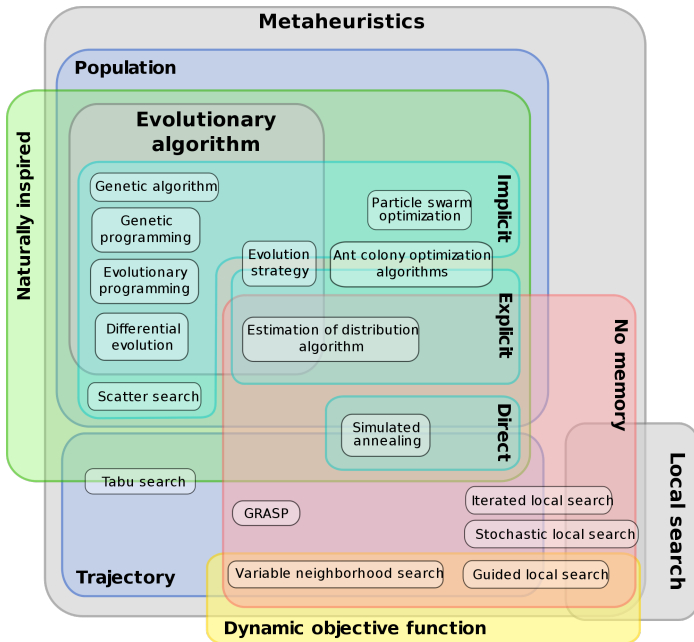
- ▶ é um processo iterativo do qual guia heurísticas subordinadas combinando conceitos inteligentes para explorar o espaço de solução

Características

- ▶ explora o espaço de busca com o intuito de encontrar soluções de boa qualidade
- ▶ é uma técnica aproximada e usualmente não-determinística
- ▶ incorpora mecanismos para evitar ficar preso em uma região do espaço de busca (ótimos locais)

Componentes de uma metaheurística





Revisão Bibliográfica

- ▶ Hybrid ALNS (2012)
- ▶ Branch-and-cut (2013)
- ▶ Branch-price-and-cut (2014)

Revisão Bibliográfica

- ▶ **Hybrid ALNS (2012)**
- ▶ Branch-and-cut (2013)
- ▶ Branch-price-and-cut (2014)

Leandro C. Coelho, Jean-François Cordeau e Gilbert Laporte. “Consistency in multi-vehicle inventory-routing”. *Em: Transportation Research Part C: Emerging Technologies* 24 (out. de 2012), pp. 270–287

Revisão Bibliográfica

- ▶ Hybrid ALNS (2012)
- ▶ **Branch-and-cut (2013)**
- ▶ Branch-price-and-cut (2014)

Leandro C Coelho e Gilbert Laporte. “The exact solution of several classes of inventory-routing problems”. Em: *Computers & Operations Research* 40.2 (fev. de 2013), pp. 558–565

Revisão Bibliográfica

- ▶ Hybrid ALNS (2012)
- ▶ Branch-and-cut (2013)
- ▶ **Branch-price-and-cut (2014)**

Guy Desaulniers, Jørgen Glomvik Rakke e Leandro C. Coelho. *A branch-price-and-cut algorithm for the inventory-routing problem*. Rel. téc. Montreal: GERAD - Department of Mathematics e Industrial Engineering Polytechnique Montréal, 2014, pp. 1–25

Algoritmo HILS-RVND

```
 $f(s^*) \leftarrow \infty$   
for  $iterMS \leftarrow 0$  until  $MaxIterMS$  do  
   $iterILS \leftarrow 0$   
   $s \leftarrow HeuristicaDeConstrucao()$   
  for  $iterILS \leftarrow 0$  until  $MaxIterILS$  do  
     $s \leftarrow BuscaLocal(s)$   
    if  $f(s) < f(s^*)$  then  
       $s^* \leftarrow s$   
       $iterILS \leftarrow -1$   
     $s \leftarrow Perturbacao(s^*)$   
     $iterILS \leftarrow iterILS + 1$   
   $iterMS \leftarrow iterMS + 1$   
 $s \leftarrow GestaoDeEstoque(s^*)$ 
```

Heurística de Construção - Fase 1

Fase 1 para período t

```
Input: Período  $t$ , Lista  $CL \leftarrow \emptyset$   
foreach  $cliente$  do  
    if  $stockout$  eh identificado then  
        AtualizarEstoque( $|stockout$  em  $t|$ )  
         $CL \leftarrow CL + \{cliente\}$   
  
Misturar( $CL$ )  
ConstruirRotas( $CL$ )  
return  $CL$ 
```

- ▶ Suprir clientes que apresentam *stockout* no período t
- ▶ Rotear os clientes atendidos

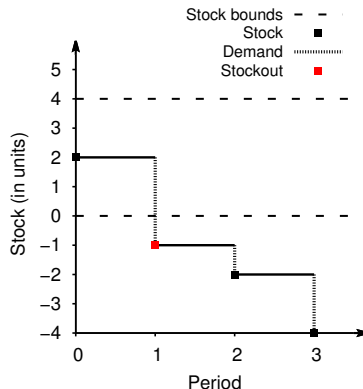
Heurística de Construção - Fase 1

Fase 1 para período t

```
Input: Período  $t$ , Lista  $CL \leftarrow \emptyset$ 
foreach cliente do
    if stockout eh identificado then
        AtualizarEstoque(|stockout em  $t$ |)
         $CL \leftarrow CL + \{\text{cliente}\}$ 
Misturar( $CL$ )
ConstruirRotas( $CL$ )
return  $CL$ 
```

Dados do estoque para cliente v_2

- ▶ Foram entregues = 0 unidades
- ▶ Períodos sem *stockout* = {1,2,3}
- ▶ $CL = \{v_1, v_3\}$



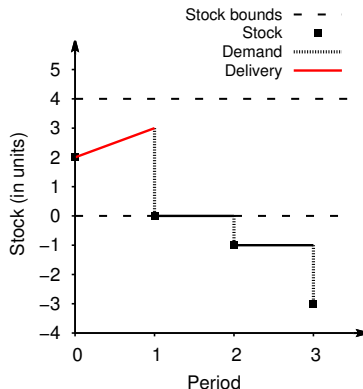
Heurística de Construção - Fase 1

Fase 1 para período t

```
Input: Período  $t$ , Lista  $CL \leftarrow \emptyset$ 
foreach cliente do
    if stockout eh identificado then
        AtualizarEstoque(|stockout em  $t$ |)
         $CL \leftarrow CL + \{\text{cliente}\}$ 
Misturar( $CL$ )
ConstruirRotas( $CL$ )
return  $CL$ 
```

Dados do estoque para cliente v_2

- ▶ Foram entregues = 1 unidade
- ▶ Períodos sem stockout = {2, 3}
- ▶ $CL = \{v_1, v_3, v_2\}$



Heurística de Construção - Fase 1

Fase 1 para período t

```
Input: Período  $t$ , Lista  $CL \leftarrow \emptyset$   
foreach cliente do  
    if stockout eh identificado then  
        AtualizarEstoque(|stockout em  $t$ |)  
         $CL \leftarrow CL + \{\textit{cliente}\}$   
  
Misturar ( $CL$ )  
ConstruirRotas ( $CL$ )  
return  $CL$ 
```



Roteamento para cliente v_2

- ▶ Custo de roteamento no período 1 = 0
- ▶ $CL = \{v_3, v_1, v_2\}$

0

Period 1

Heurística de Construção - Fase 1

Fase 1 para período t

```
Input: Período  $t$ , Lista  $CL \leftarrow \emptyset$ 
foreach cliente do
    if stockout eh identificado then
        AtualizarEstoque(|stockout em  $t$ |)
         $CL \leftarrow CL + \{cliente\}$ 
Misturar( $CL$ )
ConstruirRotas( $CL$ )
return  $CL$ 
```

Roteamento para cliente v_2

- ▶ Custo de roteamento no período 1 = 0
- ▶ $CL = \{v_3, v_1, v_2\}$

Dois critérios de inserção são usados:

1 Cliente mais próximo:

$$g(k, t) = c_{ik}^r$$

2 Cliente mais barato modificado:^a

$$g(k, t) = (c_{ik}^r + c_{kj}^r - c_{ij}^r) - \gamma(c_{0k}^r + c_{k0}^r)$$

^aUma penalidade para evitar inserção tardia de clientes localizados longe do fornecedor é usada

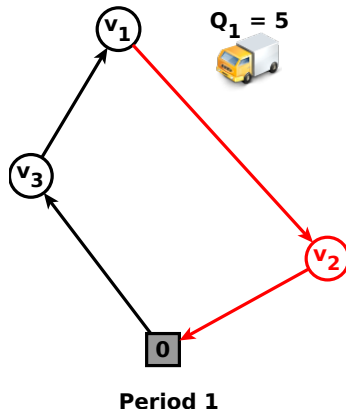
Heurística de Construção - Fase 1

Fase 1 para período t

```
Input: Período  $t$ , Lista  $CL \leftarrow \emptyset$ 
foreach cliente do
    if stockout eh identificado then
        AtualizarEstoque(|stockout em  $t$ |)
         $CL \leftarrow CL + \{\text{cliente}\}$ 
Misturar( $CL$ )
ConstruirRotas( $CL$ )
return  $CL$ 
```

Roteamento para cliente v_2

- ▶ Custo de roteamento no período 1 = 3.75
- ▶ $CL = \{v_3, v_1, v_2\}$



Heurística de Construção - Fase 2

Fase 2 para período t

```
Input: Período  $t$ , Lista  $CL$   
Misturar( $CL$ )  
foreach cliente em  $CL$  do  
     $t_f \leftarrow$  selecionar período arbitrário[ $t$ ,  $T$ ]  
    while operação é inviável do  
         $entrega \leftarrow |stockout \text{ em } t_f|$   
        ChecarViabilidade( $t_f$ ,  $entrega$ )  
        if ainda é inviável then  
             $t_f \leftarrow t_f - 1$   
        else  
            end while  
    AtualizarEstoque( $t_f$ ,  $entrega$ )  
     $s \leftarrow$  AtualizarRota( $t$ )  
return  $s$ 
```

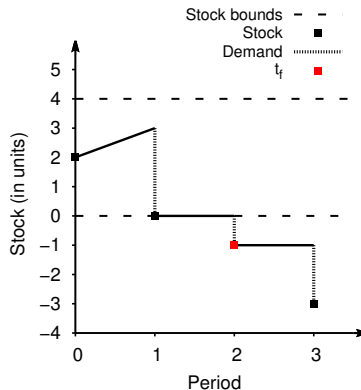
- Aumentar a quantidade de produtos entregues nos atendimentos realizados na Fase 1

Heurística de Construção - Fase 2

Fase 2 para período t

```

Input: Período  $t$ , Lista  $CL$ 
Misturar( $CL$ )
foreach cliente em  $CL$  do
     $t_f \leftarrow$  selecionar período arbitrário[ $t$ ,  $T$ ]
    while operação é inviável do
         $entrega \leftarrow |stockout \text{ em } t_f|$ 
        ChecarViabilidade( $t_f$ ,  $entrega$ )
        if ainda é inviável then
             $t_f \leftarrow t_f - 1$ 
        else
            end while
    AtualizarEstoque( $t_f$ ,  $entrega$ )
     $s \leftarrow$  AtualizarRota( $t$ )
return  $s$ 
  
```

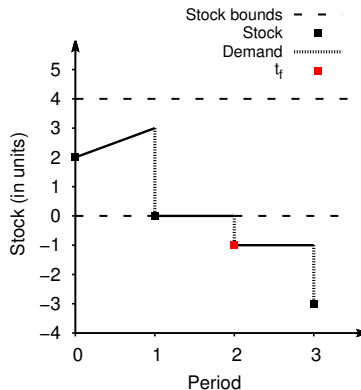


Heurística de Construção - Fase 2

Fase 2 para período t

```

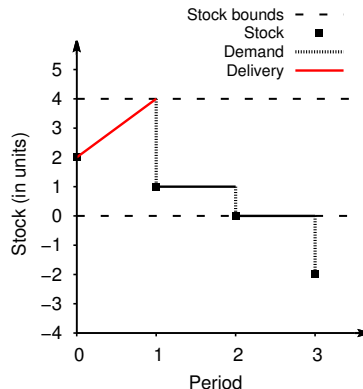
Input: Período  $t$ , Lista  $CL$ 
Misturar( $CL$ )
foreach cliente em  $CL$  do
     $t_f \leftarrow$  seleccionar período arbitrário $[t, T]$ 
    while operação é inviável do
         $entrega \leftarrow |stockout \text{ em } t_f|$ 
        ChecarViabilidade( $t_f$ ,  $entrega$ )
        if ainda é inviável then
             $t_f \leftarrow t_f - 1$ 
        else
            end while
    AtualizarEstoque( $t_f$ ,  $entrega$ )
     $s \leftarrow$  AtualizarRota( $t$ )
return  $s$ 
  
```



Heurística de Construção - Fase 2

Fase 2 para período t

```
Input: Período  $t$ , Lista  $CL$   
Misturar( $CL$ )  
foreach cliente em  $CL$  do  
     $t_f \leftarrow$  seleccionar período arbitrário $[t, T]$   
    while operação é inviável do  
         $entrega \leftarrow |stockout \text{ em } t_f|$   
        ChecarViabilidade( $t_f$ ,  $entrega$ )  
        if ainda é inviável then  
             $t_f \leftarrow t_f - 1$   
        else  
            end while  
    AtualizarEstoque( $t_f$ ,  $entrega$ )  
     $s \leftarrow$  AtualizarRota( $t$ )  
return  $s$ 
```



Heurística de Construção - Fase 2

Fase 2 para período t

Input: Período t , Lista CL

Misturar(CL)

foreach cliente em CL **do**

$t_f \leftarrow$ selecionar período arbitrário $[t, T]$

while operação é inviável **do**

$entrega \leftarrow |stockout \text{ em } t_f|$

 ChecarViabilidade(t_f , $entrega$)

if ainda é inviável **then**

$t_f \leftarrow t_f - 1$

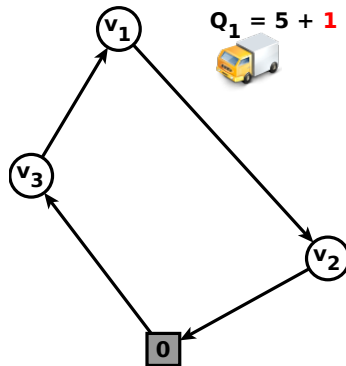
else

end while

 AtualizarEstoque(t_f , $entrega$)

$s \leftarrow$ AtualizarRota(t)

return s



Period 1

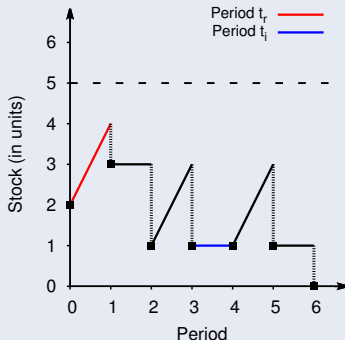
Busca Local

- ▶ Realizado por um procedimento RVND
 - ▶ *Random Variable Neighborhood Search*
- ▶ Três tipos de estruturas de vizinhança são usadas:
 - ▶ Inter-período
 - ▶ Inter-rota
 - ▶ Intrarrota

Inter-período - Service Insertion

- ▶ Realoca produtos do período t_r , adicionando um novo atendimento no período t_i
- ▶ Aumenta o custo de roteamento e reduz o custo de estocagem

Antes da aplicação



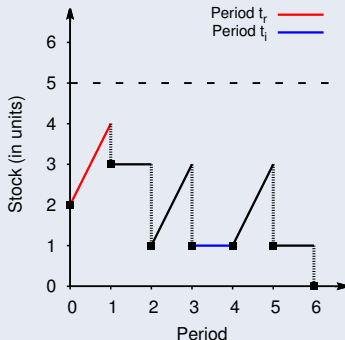
Pré-requisitos

- ▶ período $t_r < \text{período } t_i$
- ▶ um atendimento deve existir no período t_r
- ▶ nenhum atendimento deve existir no período t_i
- ▶ alguma rota com capacidade residual positiva deve existir no período t_i

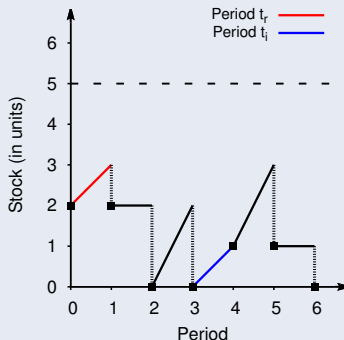
Inter-período - Service Insertion

- ▶ Realoca produtos do período t_r , adicionando um novo atendimento no período t_i
- ▶ Aumenta o custo de roteamento e reduz o custo de estocagem

Antes da aplicação



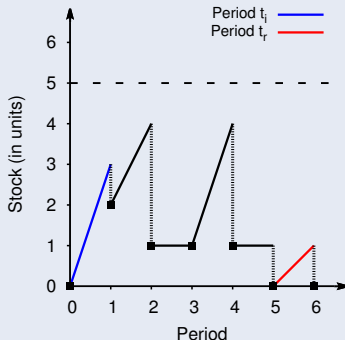
Após a aplicação



Inter-período - Service Removal

- ▶ Realoca produtos para o período t_i , removendo um atendimento existente no período t_r
- ▶ Aumenta o custo de estocagem e reduz o custo de roteamento

Antes da aplicação



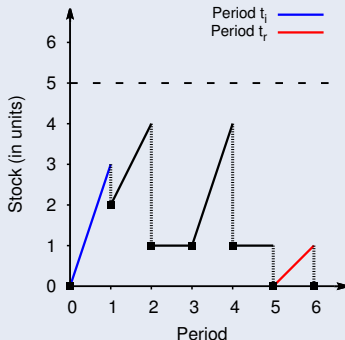
Pré-requisitos

- ▶ um atendimento deve existir nos períodos t_r e t_i
- ▶ rota associada ao cliente no período t_i , deve possuir capacidade residual suficiente para armazenar todos os produtos realocados

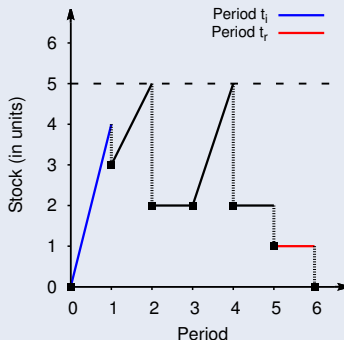
Inter-período - Service Removal

- ▶ Realoca produtos para o período t_i , removendo um atendimento existente no período t_r
- ▶ Aumenta o custo de estocagem e reduz o custo de roteamento

Antes da aplicação



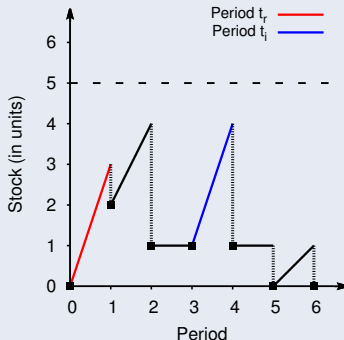
Após a aplicação



Inter-período - Shift Delivery

- ▶ Desloca parcialmente produtos de um atendimento do período **período t_r** para um novo/existente atendimento no período **período t_i**
- ▶ Apenas o custo de estocagem é reduzido

Antes da aplicação



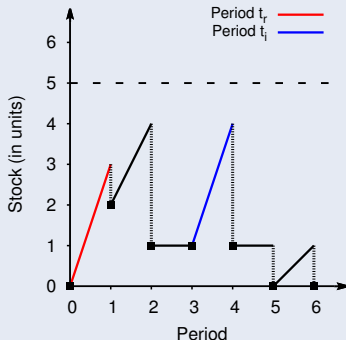
Pré-requisitos

- ▶ período $t_r < \text{período } t_i$
- ▶ um atendimento deve existir no período t_r
- ▶ alguma rota com capacidade residual positiva deve existir no período t_i

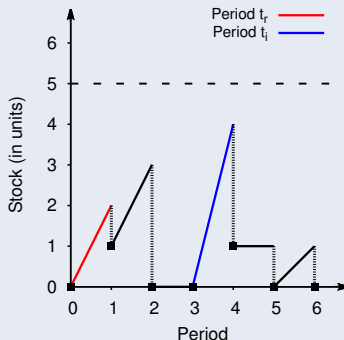
Inter-período - Shift Delivery

- ▶ Desloca parcialmente produtos de um atendimento do período **período t_r** para um novo/existente atendimento no período **período t_i**
- ▶ Apenas o custo de estocagem é reduzido

Antes da aplicação



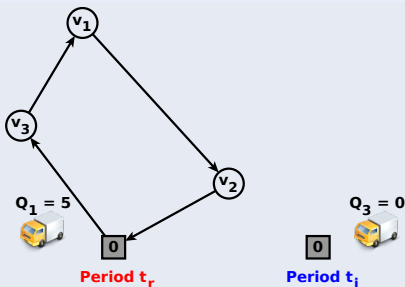
Após a aplicação



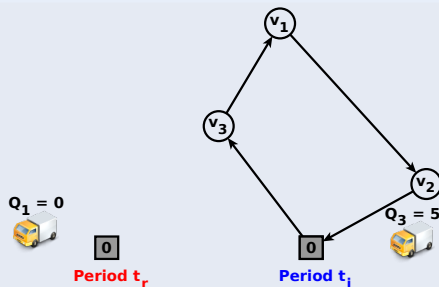
Inter-período - Swap Route

- ▶ Transfere um rota rt por completo do período t_r para o período t_i
- ▶ Custo de estocagem é reduzido

Antes da aplicação



Após a aplicação

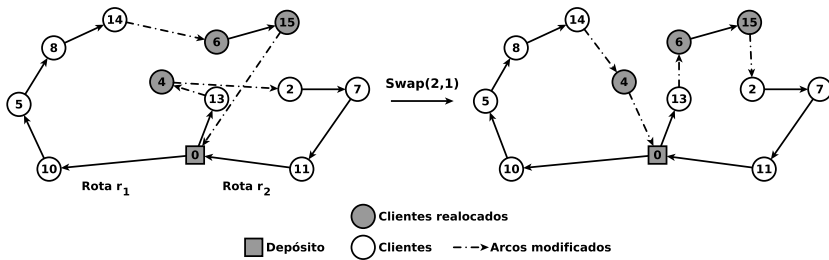


Estruturas Inter-rota

- ▶ **Swap(1,1)**: Troca dois clientes de rotas distintas
- ▶ **Swap(2,1)**: Troca um arco e um cliente de rotas distintas
- ▶ **Swap(2,2)**: Troca dois arcos de rotas distintas
- ▶ **Shift(1,0)**: Um cliente é removido e inserido em outra rota
- ▶ **Shift(2,0)**: Um arco é removido e inserido em outra rota

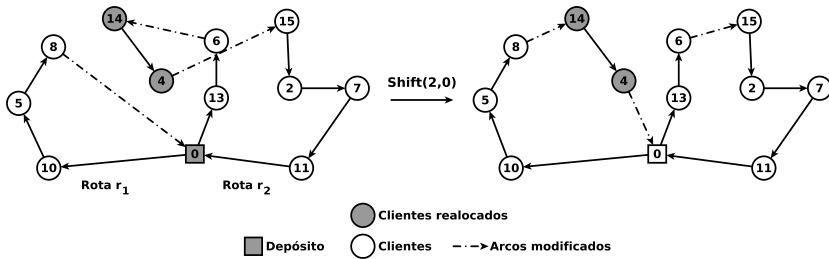
Estruturas Inter-rota

- ▶ **Swap(1,1)**: Troca dois clientes de rotas distintas
- ▶ **Swap(2,1)**: Troca um arco e um cliente de rotas distintas
- ▶ **Swap(2,2)**: Troca dois arcos de rotas distintas
- ▶ **Shift(1,0)**: Um cliente é removido e inserido em outra rota
- ▶ **Shift(2,0)**: Um arco é removido e inserido em outra rota



Estruturas Inter-rota

- ▶ **Swap(1,1)**: Troca dois clientes de rotas distintas
- ▶ **Swap(2,1)**: Troca um arco e um cliente de rotas distintas
- ▶ **Swap(2,2)**: Troca dois arcos de rotas distintas
- ▶ **Shift(1,0)**: Um cliente é removido e inserido em outra rota
- ▶ **Shift(2,0)**: Um arco é removido e inserido em outra rota

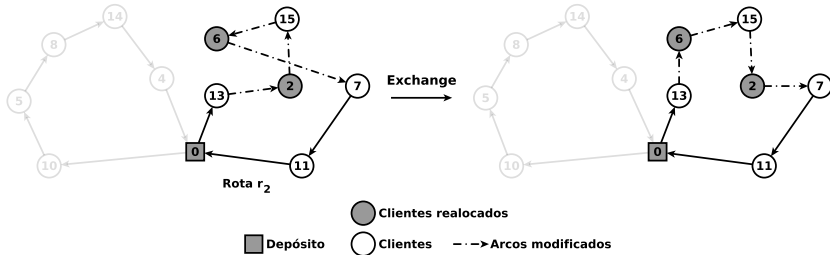


Estruturas Intrarota

- ▶ **Exchange**: Troca a posição de dois clientes
- ▶ **Reinsertion**: Um cliente é removido e reinserido em outra posição da rota
- ▶ **Or-opt2**: Dois clientes adjacentes são removidos e reinseridos em outra posição da rota
- ▶ **Or-opt3**: Três clientes adjacentes são removidos e reinseridos em outra posição da rota
- ▶ **2-opt**: Dois arcos não adjacentes (i_1, j_1) e (i_2, j_2) são removidos. Então, dois novos arcos (i_1, i_2) e (j_1, j_2) são inseridos

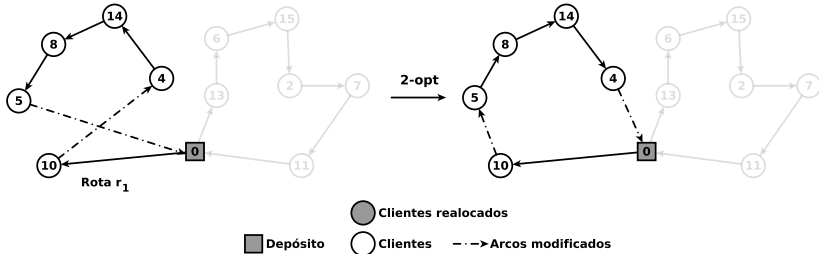
Estruturas Intrarota

- ▶ **Exchange:** Troca a posição de dois clientes
- ▶ **Reinsertion:** Um cliente é removido e reinserido em outra posição da rota
- ▶ **Or-opt2:** Dois clientes adjacentes são removidos e reinseridos em outra posição da rota
- ▶ **Or-opt3:** Três clientes adjacentes são removidos e reinseridos em outra posição da rota
- ▶ **2-opt:** Dois arcos não adjacentes (i_1, j_1) e (i_2, j_2) são removidos. Então, dois novos arcos (i_1, i_2) e (j_1, j_2) são inseridos



Estruturas Intrarota

- ▶ **Exchange:** Troca a posição de dois clientes
- ▶ **Reinsertion:** Um cliente é removido e reinserido em outra posição da rota
- ▶ **Or-opt2:** Dois clientes adjacentes são removidos e reinseridos em outra posição da rota
- ▶ **Or-opt3:** Três clientes adjacentes são removidos e reinseridos em outra posição da rota
- ▶ **2-opt:** Dois arcos não adjacentes (i_1, j_1) e (i_2, j_2) são removidos. Então, dois novos arcos (i_1, i_2) e (j_1, j_2) são inseridos



Mecanismos de perturbação

- ▶ Baseado em três estruturas inter-período:
 - ▶ ***Supply Insertion***
 - ▶ ***Supply Removal***
 - ▶ ***Shift Delivery***
- ▶ Uma das estruturas é selecionada de modo aleatório e um número arbitrário de movimentos $[1, 4]$ são executados

Gestão do estoque

Constantes

- ▶ h_i custo de estocagem para cliente i
- ▶ p^t quantidade de produto disponível no fornecedor no período t
- ▶ d_i^t demanda dos produtos no período t para cliente i
- ▶ C_i capacidade máxima de estocagem para cliente i
- ▶ Q capacidade máxima do veículo
- ▶ z_i^t assume 1, se e somente se, cliente i é atendido no período t e 0 caso contrário

Variáveis

- ▶ I_i^t quantidade de produto armazenado no período t para o cliente i
- ▶ q_i^t quantidade de produto entregue no período t para cliente i

$$\text{Minimizar} \quad \sum_{v \in V} \sum_{t \in T} h_v l_v^t \quad (1)$$

Sujeito a:

$$l_0^t = l_0^{t-1} + p^t - \sum_{v \in V'} q_v^t \quad t \in T \quad (2)$$

$$l_v^t = l_v^{t-1} + q_v^t - d_v^t \quad v \in V' \quad t \in T \quad (3)$$

$$l_v^t \geq 0 \quad v \in V \quad t \in T \quad (4)$$

$$l_v^t \leq C_v \quad v \in V' \quad t \in T \quad (5)$$

$$q_v^t \leq C_v - l_v^{t-1} \quad v \in V' \quad t \in T \quad (6)$$

$$\sum_{v \in V'} q_v^t \leq Q \quad t \in T \quad (7)$$

$$q_v^t \geq z_v^t \quad v \in V' \quad t \in T \quad (8)$$

$$q_v^t = 0 \quad v \in V' \quad t \in T, \quad \forall z_v^t = 0 \quad (9)$$

$$\text{Minimizar} \quad \sum_{v \in V} \sum_{t \in T} h_v I_v^t \quad (1)$$

Sujeito a:

$$I_0^t = I_0^{t-1} + p^t - \sum_{v \in V'} q_v^t \quad t \in T \quad (2)$$

$$I_v^t = I_v^{t-1} + q_v^t - d_v^t \quad v \in V' \quad t \in T \quad (3)$$

$$I_v^t \geq 0 \quad v \in V \quad t \in T \quad (4)$$

$$I_v^t \leq C_v \quad v \in V' \quad t \in T \quad (5)$$

$$q_v^t \leq C_v - I_v^{t-1} \quad v \in V' \quad t \in T \quad (6)$$

$$\sum_{v \in V'} q_v^t \leq Q \quad t \in T \quad (7)$$

$$q_v^t \geq z_v^t \quad v \in V' \quad t \in T \quad (8)$$

$$q_v^t = 0 \quad v \in V' \quad t \in T, \quad \forall z_v^t = 0 \quad (9)$$

Objetivo (1): minimizar custo de estocagem para a solução

$$\text{Minimizar} \quad \sum_{v \in V} \sum_{t \in T} h_v I_v^t \quad (1)$$

Sujeito a:

$$I_0^t = I_0^{t-1} + p^t - \sum_{v \in V'} q_v^t \quad t \in T \quad (2)$$

$$I_v^t = I_v^{t-1} + q_v^t - d_v^t \quad v \in V' \quad t \in T \quad (3)$$

$$I_v^t \geq 0 \quad v \in V \quad t \in T \quad (4)$$

$$I_v^t \leq C_v \quad v \in V' \quad t \in T \quad (5)$$

$$q_v^t \leq C_v - I_v^{t-1} \quad v \in V' \quad t \in T \quad (6)$$

$$\sum_{v \in V'} q_v^t \leq Q \quad t \in T \quad (7)$$

$$q_v^t \geq z_v^t \quad v \in V' \quad t \in T \quad (8)$$

$$q_v^t = 0 \quad v \in V' \quad t \in T, \quad \forall z_v^t = 0 \quad (9)$$

Restrições (2-3): estipula nível de estoque para fornecedor e clientes

$$\text{Minimizar} \quad \sum_{v \in V} \sum_{t \in T} h_v I_v^t \quad (1)$$

Sujeito a:

$$I_0^t = I_0^{t-1} + p^t - \sum_{v \in V'} q_v^t \quad t \in T \quad (2)$$

$$I_v^t = I_v^{t-1} + q_v^t - d_v^t \quad v \in V' \quad t \in T \quad (3)$$

$$I_v^t \geq 0 \quad v \in V \quad t \in T \quad (4)$$

$$I_v^t \leq C_v \quad v \in V' \quad t \in T \quad (5)$$

$$q_v^t \leq C_v - I_v^{t-1} \quad v \in V' \quad t \in T \quad (6)$$

$$\sum_{v \in V'} q_v^t \leq Q \quad t \in T \quad (7)$$

$$q_v^t \geq z_v^t \quad v \in V' \quad t \in T \quad (8)$$

$$q_v^t = 0 \quad v \in V' \quad t \in T, \quad \forall z_v^t = 0 \quad (9)$$

Restrições (4-5): define os limites para o nível de estoque

$$\text{Minimizar} \quad \sum_{v \in V} \sum_{t \in T} h_v I_v^t \quad (1)$$

Sujeito a:

$$I_0^t = I_0^{t-1} + p^t - \sum_{v \in V'} q_v^t \quad t \in T \quad (2)$$

$$I_v^t = I_v^{t-1} + q_v^t - d_v^t \quad v \in V' \quad t \in T \quad (3)$$

$$I_v^t \geq 0 \quad v \in V \quad t \in T \quad (4)$$

$$I_v^t \leq C_v \quad v \in V' \quad t \in T \quad (5)$$

$$q_v^t \leq C_v - I_v^{t-1} \quad v \in V' \quad t \in T \quad (6)$$

$$\sum_{v \in V'} q_v^t \leq Q \quad t \in T \quad (7)$$

$$q_v^t \geq z_v^t \quad v \in V' \quad t \in T \quad (8)$$

$$q_v^t = 0 \quad v \in V' \quad t \in T, \quad \forall z_v^t = 0 \quad (9)$$

Restrições (6): assegura que o nível de estoque após um serviço de entrega respeite o limite máximo de estocagem

$$\text{Minimizar} \quad \sum_{v \in V} \sum_{t \in T} h_v l_v^t \quad (1)$$

Sujeito a:

$$l_0^t = l_0^{t-1} + p^t - \sum_{v \in V'} q_v^t \quad t \in T \quad (2)$$

$$l_v^t = l_v^{t-1} + q_v^t - d_v^t \quad v \in V' \quad t \in T \quad (3)$$

$$l_v^t \geq 0 \quad v \in V \quad t \in T \quad (4)$$

$$l_v^t \leq C_v \quad v \in V' \quad t \in T \quad (5)$$

$$q_v^t \leq C_v - l_v^{t-1} \quad v \in V' \quad t \in T \quad (6)$$

$$\sum_{v \in V'} q_v^t \leq Q \quad t \in T \quad (7)$$

$$q_v^t \geq z_v^t \quad v \in V' \quad t \in T \quad (8)$$

$$q_v^t = 0 \quad v \in V' \quad t \in T, \quad \forall z_v^t = 0 \quad (9)$$

Restrições (7): garante que produtos entregues para todos os clientes atendidos pela rota rt não exceda a capacidade do veículo

$$\text{Minimizar} \quad \sum_{v \in V} \sum_{t \in T} h_v l_v^t \quad (1)$$

Sujeito a:

$$l_0^t = l_0^{t-1} + p^t - \sum_{v \in V'} q_v^t \quad t \in T \quad (2)$$

$$l_v^t = l_v^{t-1} + q_v^t - d_v^t \quad v \in V' \quad t \in T \quad (3)$$

$$l_v^t \geq 0 \quad v \in V \quad t \in T \quad (4)$$

$$l_v^t \leq C_v \quad v \in V' \quad t \in T \quad (5)$$

$$q_v^t \leq C_v - l_v^{t-1} \quad v \in V' \quad t \in T \quad (6)$$

$$\sum_{v \in V'} q_v^t \leq Q \quad t \in T \quad (7)$$

$$q_v^t \geq z_v^t \quad v \in V' \quad t \in T \quad (8)$$

$$q_v^t = 0 \quad v \in V' \quad t \in T, \quad \forall z_v^t = 0 \quad (9)$$

Restrições (8): assegura uma quantidade mínima de entrega para clientes servidos por uma rota

Conteúdo

- 1 Introdução
- 2 Logística
- 3 PERVMP
- 4 Algoritmo HILS-RVND
- 5 Resultados Computacionais**
- 6 Conclusão e trabalhos futuros

Configuração

Ambiente

- ▶ Algoritmos implementado em C++ usando compilador G++ 4.6.4
- ▶ CPLEX CONCERT TECHNOLOGY 12.2
- ▶ INTEL® CORE™2 QUAD 2.83 GHz com 8 GB
- ▶ UBUNTU 12.04 64-BIT
- ▶ Uma *thread*

Parâmetros do algoritmo

- ▶ *MaxIterMS* fixado em $(10.000 \times T \times M \times K)$
- ▶ *MaxIterILS* igual a 10
- ▶ 10 execuções
- ▶ Tempo limite de 300 segundos

Conjunto-teste de instâncias

Primeiro conjunto

- ▶ Horizonte de planejamento $H = \{3\}$
- ▶ Número de clientes $V = \{10, 15, 20, 25, 30, 35, 40, 45, 50\}$

Segundo conjunto

- ▶ Horizonte de planejamento $H = \{6\}$
 - ▶ Número de clientes $V = \{10, 15, 20, 25, 30\}$
-
- ▶ Frota homogênea de tamanho $K = \{2, 3, 4, 5\}$
 - ▶ Um único tipo de produto $M = \{1\}$
 - ▶ Custos de estocagem baixo e alto
 - ▶ 560 problemas-teste (Coelho, 2012)

Resultados computacionais - Parte 1

- ▶ *Hybrid ALNS* (2013)
- ▶ Apenas um subconjunto dos problemas-teste foram utilizados (Coelho, 2013)

Ambiente (Coelho, 2013)

- ▶ Dual Core AMD OPTERON™ 2.20 GHz com 12 GB
- ▶ CPLEX CONCERT TECHNOLOGY 12.2
- ▶ Uma *thread*

Resultados computacionais

	HILS-RVND				ALNS
	Quant	Quant (%)	Avg Gap ALNS (%)	Avg Time (s)	Time (s)
Best	32	66,66	-0,76	-	-
Draw	0	0,00	0,00	-	-
Worst	16	33,33	0,51	-	-
TOTAL	48	100,00	-0,33	77,40	4.934,69

Tabela : Resumo dos resultados para HILS-RVND

Resultados computacionais - Parte 2

- ▶ *Branch-and-cut* (2013)
- ▶ *Branch-price-and-cut* (2014)

Ambiente (Coelho, 2014)

- ▶ INTEL® CORE™i7 3.40 GHZ com 16 GB
- ▶ CPLEX CONCERT TECHNOLOGY 12.2
- ▶ Uma *thread* com tempo limite de 7200 segundos

Resultados computacionais

	HILS-RVND					Exact
	Quant	Quant (%)	Opt	Avg Gap (%)	Avg Time (s)	Time (s)
Best	125	22,32	-	-8,99	-	-
Draw	127	22,68	124	0,00	-	-
Worst	308	55,00	-	2,22	-	-
TOTAL	560	100,00	124	-0,76	104,19	3.598,10

Tabela : Resumo dos resultados para HILS-RVND

Resultados computacionais

	HILS-RVND					Exact
	Quant	Quant (%)	Opt	Avg Gap (%)	Avg Time (s)	Time (s)
Best	61	21,79	-	-11,35	-	-
Draw	76	27,14	74	0,00	-	-
Worst	143	51,07	-	3,30	-	-
TOTAL	280	100,00	74	-0,74	102,90	3.629,02

Tabela : Resumo dos resultados para problemas-teste com **custos de estocagem baixo**

Resultados computacionais

	HILS-RVND					Exact
	Quant	Quant (%)	Opt	Avg Gap (%)	Avg Time (s)	Time (s)
Best	64	22,86	-	-6,75	-	-
Draw	51	18,21	50	0,00	-	-
Worst	165	58,93	-	1,29	-	-
TOTAL	280	100,00	50	-0,77	105,48	3.567,18

Tabela : Resumo dos resultados para problemas-teste com **custos de estocagem alto**

Conteúdo

- 1 Introdução
- 2 Logística
- 3 PERVMP
- 4 Algoritmo HILS-RVND
- 5 Resultados Computacionais
- 6 Conclusão e trabalhos futuros**

Conclusão e trabalhos futuros

Conclusão

- ▶ Processos de negócios podem ter seus custos operacionais reduzidos de modo considerável
- ▶ Metaheurísticas híbridas são boas alternativas para problemas de grande porte
- ▶ Podem ser adaptadas para operar em casos reais (*e.g.* Petrobras e Liquigás)

Trabalhos futuros

- ✓ Estender o algoritmo para trabalhar com outras variantes
- 🕒 Desenvolver um *framework* para facilitar o gerenciamento de processos de negócios
- 🕒 Incorporar incerteza/imprevisibilidade ao problema

Otimização de Processos de Negócios em Cadeia de Suprimentos

Edcarllos Santos

Email:

esantos@ic.uff.br

Universidade Federal Fluminense
Instituto de Computação
Niterói - RJ

Outubro/2016

