

Um Algoritmo Branch-and-price para o Problema de Roteamento de Veículos com *Cross-Docking* para Frotas Heterogêneas

Fernando Afonso Santos^{1,2}

¹Universidade Federal de Itajubá – Campus Itabira
Rua São Paulo, 377 - Bairro Amazonas - Itabira - MG, CEP 35900-373
fernandosantos@unifei.edu.br

Geraldo Robson Mateus²
Alexandre Salles da Cunha²

²Universidade Federal de Minas Gerais
Av. Antônio Carlos, 6627 - Pampulha - Belo Horizonte - MG CEP 31270-901
{mateus, acunha}@dcc.ufmg.br

RESUMO

O Problema de Roteamento de Veículos com *Cross-Docking* (PRVCD) surge da integração do problema clássico de roteamento de veículos com o escalonamento de mercadorias no armazém de *Cross-Docking*. Este artigo descreve uma formulação matemática de geração de colunas e um algoritmo branch-and-price para o PRVCD para frotas heterogêneas. Os resultados do algoritmo branch-and-price são comparados a resultados obtidos através da solução de um modelo de fluxos, que foi dominado para todas as instâncias testadas.

PALAVRAS CHAVE. Problema de Roteamento de Veículos, *Cross-Docking*, Algoritmo Branch-and-price

ABSTRACT

The Vehicle Routing Problem with Cross-Docking (VRPCD) arises by integrating the classical vehicle routing problem with the scheduling of merchandizes at Cross-Docking warehouse. This paper describes a column generation formulation and a branch-and-price algorithm for the VRPCD for heterogeneous fleets. The algorithm results are faced with those obtained by solving a network flow model and the branch-and-price algorithm results dominates the solutions from network flow model for all instances of our test set.

KEYWORDS. Vehicle Routing Problem, Cross-Docking, Branch-and-price Algorithm

1. Introdução

Diferentes alternativas vêm sendo estudadas para auxiliar no processo de distribuição de mercadorias, uma vez que custos de transporte e estocagem podem representar até 30% do valor de uma mercadoria ao consumidor [King e Mast (1997)]. Em particular, para os sistemas de cadeias de suprimentos onde mercadorias devem ser coletadas em um conjunto de fornecedores para satisfazer as demandas de um conjunto de consumidores, foram propostos os armazéns de *Cross-Docking* [Apte e Viswanathan (2000)], que permitem organizar a distribuição e aumentar a eficiência na entrega das mercadorias.

As mercadorias coletadas nos fornecedores são levadas ao armazém de *Cross-Docking*, onde passam por um processo de consolidação antes de serem entregues aos respectivos consumidores. Na consolidação, diferentes tarefas inerentes à distribuição podem ser executadas, mas principalmente as mercadorias podem ser rearranjadas entre os veículos de modo a facilitar sua entrega.

Para solucionar o problema da distribuição das mercadorias usando armazéns de *Cross-Docking* é necessário considerar além do roteamento de veículos, também o escalonamento das mercadorias no *Cross-Docking*, determinando se haverá trocas e entre quais veículos. Desta forma, surge um novo problema de otimização, o Problema de Roteamento de Veículos com *Cross-Docking* (PRVCD).

O PRVCD pode ser definido em um grafo $G = (V, A)$ onde $V = \{\{0\}, F, C\}$, $F = \{1, \dots, n\}$ denota os fornecedores e $C = \{1', \dots, n'\}$ os consumidores do problema, além de 0 que representa o armazém de *Cross-Docking*. O conjunto de arestas A é dividido em duas partições A_F e A_C tais que $A = A_F \cup A_C$ ($A_F \cap A_C = \emptyset$), onde $A_F = \{(i, j) : i, j \in \{0, 1, \dots, n\}\}$ representa o conjunto de arcos que conecta os fornecedores entre si e ao *Cross-Docking* e em $A_C = \{(i, j) : i, j \in \{0, 1', \dots, n'\}\}$ estão os arcos conectando os consumidores entre si e o *Cross-Docking*. Considere também que $P = \{p_i \leftarrow (i, i', q_i) : i = 1, \dots, n\}$ denote um conjunto de requisições a serem atendidas, onde cada requisição é dada por uma carga não divisível $q_i > 0$ que deve ser coletada no fornecedor i e entregue no consumidor i' . Nesta definição considera-se que $|P| = |F| = |C|$, mas é possível transformar problemas de distribuição com conjuntos F e C de diferentes cardinalidades no problema acima usado apenas transformações no grafo.

Para satisfazer as requisições de P tem-se disponível uma frota heterogênea de veículos K de capacidade Q^k , $\forall k \in K$. Assumindo que a cada arco do grafo sejam atribuídos custos em função do veículo que o atravessa $\{c_{ij}^k \geq 0 : (i, j) \in A, k \in K\}$, o PRVCD consiste em definir $|K|$ rotas nas quais os veículos deixam o *Cross-Docking*, visitam fornecedores para coletar mercadorias e retornam ao *Cross-Docking*, onde ocorre a etapa de consolidação e mercadorias podem eventualmente trocar de veículos para facilitar sua entrega. Após isto, são necessárias outras $|K|$ rotas nas quais os veículos deixam o *Cross-Docking* para entregar as mercadorias aos respectivos consumidores, retornando finalmente ao *Cross-Docking*. Além dos custos de roteamento são considerados também custos para trocar mercadorias no *Cross-Docking*, assim, caso alguma mercadoria q_i seja carregada/descarregada pelo veículo k no *Cross-Docking*, um custo $c_i^k \geq 0$ será incluído na solução.

Portanto, no PRVCD procura-se por uma atribuição de rotas e trocas de mercadorias no *Cross-Docking* que minimize o custo da solução. A figura 1 ilustra um exemplo de PRVCD onde $K = \{k_1, k_2\}$, $F = \{1, 2, 3, 4, 5\}$ e $C = \{1', 2', 3', 4', 5'\}$. Observe que a carga das requisições p_1 e p_2 foi trocada do veículo k_1 para k_2 , enquanto a carga da requisição p_4 de k_2 para k_1 . Neste caso, além dos custos de roteamento seriam incluídos na solução os custos associados à troca de tais mercadorias.

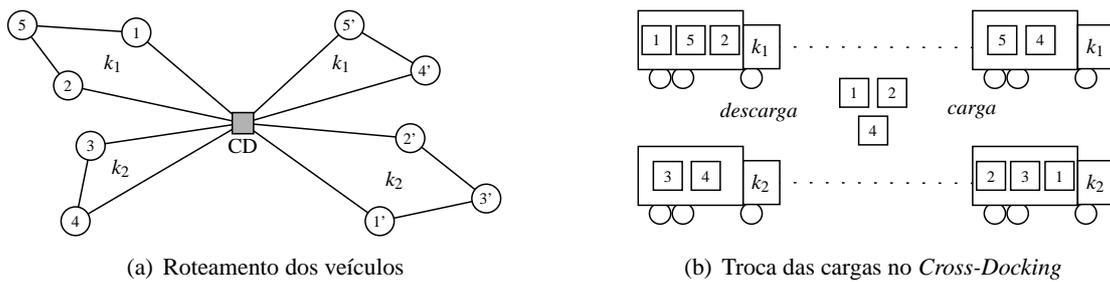


Figura 1: Exemplo de solução do PRVCD

O grande benefício apresentado ao utilizar armazéns de *Cross-Docking* na distribuição de mercadorias é a dinamicidade com a qual elas passam pelos centros de distribuição. Desta forma, é possível usufruir de toda a estrutura disponível nestas unidades sem a necessidade de custos de estocagem, que contribuem com uma parcela considerável para o aumento nos custos de distribuição. Por outro lado, uma grande dificuldade enfrentada ao lidar com os armazéns de *Cross-Docking* é a necessidade de sincronização e organização da cadeia de suprimentos, uma vez que a indisponibilidade ou o excesso de um determinado produto pode trazer problemas aos envolvidos, sejam fornecedores, consumidores ou o próprio *Cross-Docking*.

Devido à sua grande eficiência na distribuição de mercadorias, diversas companhias de mercado global vêm adotando os armazéns de *Cross-Docking*. Entre elas Wal Mart [Gue (2001)], UPS [Forger (1995)] e Toyota [Witt (1998)] são apenas algumas que permitiram a publicação dos resultados de seus sistemas de distribuição após a utilização dos armazéns de *Cross-Docking*.

Estudos a respeito dos armazéns de *Cross-Docking* podem ser encontrados em diferentes níveis, como no estratégico onde são verificados requisitos para a instalação de armazéns em uma determinada cadeia de suprimentos ou no nível de planejamento para determinar sua localização, seu formato e quantidade de docas, por exemplo. No entanto, a maioria dos trabalhos disponíveis na literatura procuram por soluções envolvendo o nível operacional dos *Cross-Docking*, buscando por alternativas que facilitem e minimizem os custos de suas operações.

Neste contexto se destacam os trabalhos de Yu e Egbelu (2008) e Boysen (2010), que tratam de maneira específica do escalonamento das mercadorias no *Cross-Docking* de modo a minimizar tempo/custo. No entanto, as rotas utilizadas pelos veículos para coletar e entregar as mercadorias não são consideradas, os autores se limitam apenas a determinar as atribuições internas no *Cross-Docking*.

Procurando por uma abordagem mais completa do sistema de distribuição com *Cross-Docking* foi proposto o PRVCD por Lee et al. (2006). Para a sua solução os autores assumem que existe apenas um armazém de *Cross-Docking* e que existem janelas de tempo nos fornecedores e consumidores, sendo que o horizonte de planejamento é realizado para um dia de trabalho. É apresentada uma formulação matemática para o problema e uma heurística Busca Tabu para obter soluções aproximadas. A heurística apresentou qualidade nos resultados, com soluções em média a 5% das soluções ótimas, no entanto, apenas instâncias de pequeno porte foram consideradas.

Um PRVCD com características semelhantes ao trabalho de Lee et al. (2006) foi estudado por Wen et al. (2008), incluindo algumas características operacionais de uma em-

presa de logística que utiliza armazéns de *Cross-Docking*. Novamente uma formulação matemática foi apresentada, além de uma heurística Busca Tabu para aproximações com qualidade. Testes foram executados em grupos de instâncias derivadas de problemas reais com até 200 requisições e as soluções obtidas pela heurística ficaram em média a menos de 5% da solução ótima.

Os trabalhos de Lee et al. (2006) e Wen et al. (2008) foram os primeiros a lidar com o PRVCD, no entanto, em ambos os custos para trocar mercadorias no *Cross-Docking* foram negligenciados. Neste sentido, os trabalhos de Santos et al. (2010), Santos et al. (2011b) e Santos et al. (2011a) foram propostos para tratar o PRVCD quando a troca de mercadorias implica em custos na solução do problema. Em [Santos et al. (2010)] foram propostas duas formulações matemáticas baseadas em fluxos em redes para o PRVCD com custos de troca e resultados computacionais foram obtidos usando o *solver* programação inteira do CPLEX. Devido à solução do PRVCD ter se mostrado ineficiente usando modelos de fluxos em redes, uma formulação de geração de colunas e um algoritmo branch-and-price para solucioná-la foram propostos em [Santos et al. (2011b)]. Os resultados do algoritmo branch-and-price dominaram aqueles obtidos pelo CPLEX com a formulação de fluxos, mas devido à simetria na formulação de geração de colunas os resultados ainda foram modestos, permitindo calcular soluções ótimas apenas problemas com até 15 requisições. Buscando por alternativas para solucionar o PRVCD com custos de troca, Santos et al. (2011a) apresentaram um novo modelo de geração de colunas e um algoritmo branch-and-price que permite avaliar soluções ótimas para problemas maiores, de até 20 requisições, obtendo ainda gaps de dualidade abaixo de 1% para instâncias com 25 e 30 requisições (50 e 60 vértices, respectivamente).

Todos os trabalhos acerca do PRVCD descritos acima lidam com frotas homogêneas, o que constitui uma limitação para a sua aplicação em contextos reais de distribuição de mercadorias. Desta forma, será apresentada neste artigo uma variação do trabalho proposto por [Santos et al. (2011b)], onde a formulação de geração de colunas e o algoritmo branch-and-price serão generalizados para contemplar também frotas heterogêneas.

2. Formulação matemática

Seja R o conjunto de todas as rotas elementares e viáveis em capacidade para visitar os vértices fornecedores e analogamente R' o conjunto de rotas para os consumidores. A formulação matemática consiste em atribuir uma variável de decisão binária para cada rota e cada veículo, determinando se a rota foi percorrida pelo veículo ou não, tanto para fornecedores quanto para consumidores. Além disto, um outro conjunto de variáveis de decisão deve ser usado para determinar se houve ou não troca de mercadorias entre os veículos. Desta forma, considere as seguintes variáveis de decisão:

$$\begin{aligned} \lambda_r^k &= \begin{cases} 1 - \text{A rota } r \text{ será percorrida pelo veculo } k \text{ no roteamento dos fornecedores} \\ 0 - \text{Caso contrário} \end{cases} \\ \gamma_{r'}^k &= \begin{cases} 1 - \text{A rota } r' \text{ será percorrida pelo veculo } k \text{ no roteamento dos consumidores} \\ 0 - \text{Caso contrário} \end{cases} \\ \tau_i^k &= \begin{cases} 1 - \text{O veculo } k \text{ carrega/descarrega a mercadoria } q_i \text{ no Cross-Docking} \\ 0 - \text{Caso contrário} \end{cases} \end{aligned}$$

Além das variáveis de decisão, são utilizados três parâmetros: a_r^i assume valor 1 se o fornecedor $i \in F$ é visitado pela rota $r \in R$ ou 0 caso contrário, $b_{r'}^{i'}$ indicando se o consumidor $i' \in C$ é ou não visitado pela rota $r' \in R'$, além de $c_r^k = \sum_{(i,j) \in r} c_{ij}^k$ que denota o custo para

o veículo k atravessar os arcos da rota r . Segue abaixo um modelo de programação linear inteira baseado em rotas para o PRVCD.

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

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

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

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

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

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

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

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

A função objetivo (1) busca minimizar a soma dos custos de roteamento com as rotas usadas para visitar os fornecedores e consumidores, além dos custos de troca, incluídos quando alguma mercadoria é carregada/descarregada por um veículo no *Cross-Docking*. Quanto às restrições, as equações (2) e (3) asseguram que todos os veículos serão utilizados tanto para coletar mercadorias nos fornecedores quanto entregá-las nos consumidores, enquanto que (4) e (5) garantem que cada vértice no conjunto dos fornecedores (e consumidores) seja visitado exatamente uma vez. Por fim, as desigualdades (6) e (7) asseguram que, caso algum veículo carregue ou descarregue uma mercadoria no *Cross-Docking*, a variável de troca associada assumirá valor 1. Em (8) estão definidos os espaços das variáveis de decisão.

2.1. Geração de Colunas

A cardinalidade dos conjuntos R e R' cresce exponencialmente com o aumento do número de vértices, levando a um número intratável de variáveis no modelo (1)-(8), mesmo para problemas de pequeno porte. Uma alternativa para solucionar o modelo é considerar os subconjuntos $S \subset R$ e $S' \subset R'$ com um número reduzido de rotas e inserindo dinamicamente rotas para minimizar (1) até que tais rotas não existam. Para realizar a precificação de rotas reescreve-se o modelo (1)-(8) em função dos conjuntos S e S' e relaxa-se a integralidade das variáveis, obtendo-se o Programa Linear Master Restrito (PLMR). Assim é possível utilizar informações duais do PLMR na geração de colunas (precificação de rotas). A formulação do PLMR segue.

$$w_{GC} = \min \sum_{r \in S} c_r^k \sum_{k \in K} \lambda_r^k + \sum_{r' \in S'} c_{r'}^k \sum_{k \in K} \gamma_{r'}^k + \sum_{k \in K} \sum_{p_i \in P} c_i^k \tau_i^k \quad (9)$$

$$\sum_{r \in S} \lambda_r^k = 1 \quad \forall k \in K \quad (10)$$

$$\sum_{r' \in S'} \gamma_{r'}^k = 1 \quad \forall k \in K \quad (11)$$

$$\sum_{r \in S} a_r^i \sum_{k \in K} \lambda_r^k = 1 \quad \forall i \in F \quad (12)$$

$$\sum_{r' \in S'} b_{r'}^{i'} \sum_{k \in K} \gamma_{r'}^k = 1 \quad \forall i' \in C \quad (13)$$

$$\sum_{r \in S} \lambda_r^k a_r^i - \sum_{r' \in S'} \gamma_{r'}^k b_{r'}^{i'} + \tau_i^k \geq 0 \quad \forall p_i \in P, \forall k \in K \quad (14)$$

$$-\sum_{r \in S} \lambda_r^k a_r^i + \sum_{r' \in S'} \gamma_{r'}^k b_{r'}^{i'} + \tau_i^k \geq 0 \quad \forall p_i \in P, \forall k \in K \quad (15)$$

$$\lambda \in \mathbb{R}^{|S| \times |K|}, \gamma \in \mathbb{R}^{|S'| \times |K|}, \tau \in \mathbb{R}^{|P| \times |K|} \quad (16)$$

Ao associar as variáveis duais $\{\alpha^k : k \in K\}$, $\{\beta^k : k \in K\}$, $\{\theta_i : i \in F\}$, $\{\mu_{i'} : i' \in C\}$, $\{\pi_i^k : k \in K, p_i \in P\}$ e $\{\chi_i^k : k \in K, p_i \in P\}$ respectivamente às restrições (10)-(15) a seguinte formulação dual é obtida:

$$\max \sum_{k \in K} \alpha^k + \sum_{k \in K} \beta^k + \sum_{i \in F} \theta_i + \sum_{i' \in C} \mu_{i'} \quad (17)$$

$$\alpha^k + \sum_{i \in F} a_r^i \theta_i + \sum_{i \in F} a_r^i \pi_i^k - \sum_{i \in F} a_r^i \chi_i^k \leq c_r^k \quad \forall r \in S, \forall k \in K \quad (18)$$

$$\beta^k + \sum_{i' \in C} b_{r'}^{i'} \mu_{i'} - \sum_{i' \in C} b_{r'}^{i'} \pi_i^k + \sum_{i' \in C} b_{r'}^{i'} \chi_i^k \leq c_{r'}^k \quad \forall r' \in S', \forall k \in K \quad (19)$$

$$\pi_i^k + \chi_i^k \leq c_i^k \quad \forall k \in K, \forall p_i \in P \quad (20)$$

$$\alpha \in \mathbb{R}^{|K|}, \beta \in \mathbb{R}^{|K|}, \theta_i \in \mathbb{R}^{|F|}, \mu_{i'} \in \mathbb{R}^{|C|} \quad (21)$$

$$\pi_i^k \in \mathbb{R}_+^{|K| \times |P|}, \chi_i^k \in \mathbb{R}_+^{|K| \times |P|} \quad (22)$$

A tarefa de precificar colunas para calcular o limite de relaxação linear w_{GC} do PLMR (modelo (9)-(16)) consiste em procurar por rotas $r \in R$ e $r' \in R'$ que violem respectivamente as restrições (18) e (19) do modelo dual. Ao incluir os custos duais das variáveis θ, μ, π e χ nos respectivos fornecedores e consumidores, o subproblema de geração de colunas pode ser formulado através do Problema de Caminho Mínimo Elementar com Restrições de Recurso (PCMERR). Em nosso caso, o único recurso a ser considerado é a capacidade dos veículos, uma vez que as janelas de tempo não foram consideradas nesta versão do problema. Para solucionar o PCMERR foi implementado o algoritmo de programação dinâmica proposto por Feillet et al. (2004). Caso o custo da rota retornada pelo PCMERR viole a restrição (18) (ou (19) se calculada para consumidores), esta deve ser inserida no PLMR e uma nova iteração começa. Caso contrário, se não existirem rotas violadas para nenhum veículo k tanto para fornecedores quanto para consumidores, a geração de colunas para e o limite de relaxação linear do PLMR é w_{GC} .

3. Algoritmo Branch-and-price

Caso a solução do PLMR seja inteira, o modelo (1)-(8) está resolvido e seu valor é dado por w_{GC} . Caso contrário, deve-se recorrer à estratégia de branching. A regra de branching foi definida em função das variáveis τ_i^k por apresentarem um maior impacto na estrutura do PLMR. A escolha do par p_i, k que defina em qual variável τ_i^k deve ser feito branching em uma dada iteração é feita avaliando-se a incerteza (grau de fractionalidade) das variáveis de rota para o veículo k que coletam e entregam a requisição p_i . Deste modo, calcula-se a estimativa

$$\psi_i^k := \sum_{r \in S} \min\{a_r^i \lambda_r^k, 1 - a_r^i \lambda_r^k\} + \sum_{r' \in S'} \min\{b_{r'}^i \gamma_{r'}^k, 1 - b_{r'}^i \gamma_{r'}^k\}$$

e aquela variável ψ_i^k com maior índice de fractionalidade definirá a variável τ_i^k a realizar o branching na iteração.

Note, contudo, que o branching em variáveis apenas fixando $\tau_i^k = 0$ e $\tau_i^k = 1$ não é suficiente para assegurar integralidade às soluções do PLMR. Isto, pois $\tau_i^k = 0$ pode significar que no PRVCD, tanto o veículo k coletou e entregou a mercadoria da requisição p_i , quanto não coletou nem entregou. Da mesma forma para $\tau_i^k = 1$ que representa que houve uma troca no *Cross-Docking*, mas não especifica se o veículo k coletou a mercadoria q_i e não entregou ou se ele a entregou sem ter coletado. Por este motivo, foi implementado um branching em restrições onde um nó dará origem a outros 4, de acordo com as seguintes regras

(i)	(ii)	(iii)	(iv)
$\tau_i^k = 0$	$\tau_i^k = 0$	$\tau_i^k = 1$	$\tau_i^k = 1$
$\sum_{r \in R} a_r^i \lambda_r^k = 1$	$\sum_{r \in R} a_r^i \lambda_r^k = 0$	$\sum_{r \in R} a_r^i \lambda_r^k = 1$	$\sum_{r \in R} a_r^i \lambda_r^k = 0$
$\sum_{r' \in R'} b_{r'}^i \gamma_{r'}^k = 1$	$\sum_{r' \in R'} b_{r'}^i \gamma_{r'}^k = 0$	$\sum_{r' \in R'} b_{r'}^i \gamma_{r'}^k = 0$	$\sum_{r' \in R'} b_{r'}^i \gamma_{r'}^k = 1$

De acordo com as regras acima, cada nó ao ser criado carregará consigo as informações de como a mercadoria q_i será tratada pelo veículo k . Os nós (i) e (ii) asseguram que a mercadoria q_i não será carregada/descarregada no *Cross-Docking*, sendo que ela pode ser coletada e entregue por k (nó (i)) ou nem coletada nem entregue por k (nó (ii)). O nó (iii) determina que haverá a descarga da mercadoria q_i no *Cross-Docking*, uma vez que o veículo k coleta esta mercadoria e não a entrega, enquanto que pelo nó (iv) a mercadoria q_i será carregada pelo veículo k no *Cross-Docking*, uma vez que este veículo a entregará sem ter coletado. Estas restrições de branching combinadas às demais restrições do problema asseguram a integralidade das variáveis ao longo da árvore de branching. A investigação dos nós nesta árvore é feita usando a estratégia de *best-bound*.

A inicialização do PLMR é feita ao definir $|K|$ rotas aleatórias (elementares e viáveis em capacidade) para o conjunto S (fornecedores) e outras $|K|$ para o conjunto S' (consumidores). Também foi avaliada uma heurística para o PRV com o objetivo de gerar um *pool* de rotas com maior qualidade para os conjuntos S e S' , no entanto, o tempo para se alcançar o nó raiz do PLMR não se alterou. Portanto, foram mantidas apenas as rotas aleatórias.

Assim que concluída a geração de colunas para o nó raiz do PLMR, uma heurística de geração de colunas é chamada. Esta heurística consiste em substituir os conjuntos R e R' do modelo (1)-(8) respectivamente por S e S' e usar o pacote CPLEX para solucionar o problema de otimização inteira resultante. O resultado da solução encontrada pelo CPLEX consiste em um limite primal para o problema.

4. Resultados Computacionais

Nesta seção são mostrados os resultados de experimentos computacionais realizados para avaliar o comportamento do algoritmo branch-and-price implementado. Para a execução dos experimentos, foi usado um conjunto de instâncias baseadas em problemas reais de uma empresa logística dinamarquesa disponíveis por Santos et al. (2011a), adaptadas de Wen et al. (2008). Neste conjunto há instâncias com $|P| = \{10, 15, 20, 25, 30\}$ onde para cada quantidade de requisições existem 5 instâncias $\{a, b, c, d, e\}$. Os custos dos arcos são calculados de acordo com a sua distância euclidiana e os custos de troca no *Cross-Docking* são definidos como $c_i^k = 15, \forall k \in K, \forall p_i \in P$ para todas as instâncias, que estão disponíveis para download em <http://www.dcc.ufmg.br/~fsantos/instances/>.

Como não existem instâncias para o PRVCD para frotas heterogêneas, as instâncias para frotas homogêneas acima tiveram que ser adaptadas. Seja $\{c_{ij} : (i, j) \in A\}$ o conjunto dos custos dos arcos e Q a capacidade de qualquer veículo na instância homogênea. Para a concepção de um conjunto de instâncias heterogêneas foi introduzido o fator $\Theta_k \in [0.5, 1.5]$ que definirá para cada veículo $k \in K$ a sua escala em relação aos veículos das instâncias homogêneas. Assumindo que sejam atribuídos índices $1, \dots, |K|$ aos veículos do conjunto K e que $i(k)$ seja uma função que retorne este índice para $k \in K$, o fator de escala para os veículos heterogêneos será dado por

$$\Theta_k = 0.5 + \frac{i(k) - 1}{|K| - 1}.$$

Após calculados os fatores são definidos os custos $\{\Theta_k c_{ij}^k : (i, j) \in A, k \in K\}$ e as capacidades dos veículos $Q^k = \lceil \Theta_k Q \rceil$ nas instâncias heterogêneas. Note que, nas instâncias heterogêneas consideradas neste trabalho, todos os veículos da frota são distintos, mas outras funções para calcular Θ_k podem ser obtidas para contemplar diferentes frotas heterogêneas. Além do mais, caso seja feita uma atribuição $\Theta_k = f, \forall k \in K$ onde $f \in \mathbb{R}_+$, o grupo de instâncias resultante será homogêneo.

Os algoritmos discutidos nesta seção foram codificados em C++, usando o pacote de otimização CPLEX versão 12 como *solver* de problemas lineares e inteiros. Os experimentos computacionais foram executados em uma máquina Intel Core 2 Quad de 2.2 GHz e 4 Gigabytes de memória RAM.

O primeiro experimento será conduzido para avaliar o desempenho do algoritmo branch-and-price descrito nas seções 2 e 3. Para isto, ele será comparado a uma formulação de fluxos em rede para o PRVCD com frota heterogênea adaptada de Santos et al. (2010). Nesta formulação, são consideradas $|K|$ *commodities*, uma por veículo, sendo solucionada pelo CPLEX explorando o recurso de paralelismo com até 4 *threads* simultâneas. Um limite de tempo de três horas foi imposto para a execução dos algoritmos. Os resultados na Tabela 1 apresentam nas colunas 1, 2 e 3 a descrição da instância, a seguir são apresentados o melhor limite dual e o melhor limite primal obtidos pelo algoritmo branch-and-price, rotulados respectivamente como MLD e MLP, além do tempo gasto pelo algoritmo para solucionar a instância e o gap de dualidade existente quando o limite de tempo for atingido, calculado como $\frac{MLP - MLD}{MLP}$. Entradas similares estão nas próximas 4 colunas considerando os resultados obtidos pelo CPLEX ao solucionar a formulação de fluxos em rede.

O algoritmo branch-and-price encontra a solução ótima para 12 das 25 instâncias testadas, enquanto para aquelas instâncias nas quais o limite de tempo é atingido o gap de dualidade médio é de 4.7%. Por outro lado a solução do modelo de fluxos, mesmo executando

P	K	instância	Algoritmo Branch-and-price				Modelo de Fluxos			
			MLD	MLP	tempo	GAP	MLD	MLP	tempo	GAP
10	4	a	1677.1	1677.1	6s	0.0%	1677.1	1677.1	30s	0.0%
		b	1712.4	1712.4	99s	0.0%	1712.4	1712.4	617s	0.0%
		c	1975.7	1975.7	48s	0.0%	1975.7	1975.7	166s	0.0%
		d	1647.9	1647.9	10s	0.0%	1647.9	1647.9	26s	0.0%
		e	2027.9	2027.9	6s	0.0%	2027.9	2027.9	30s	0.0%
15	6	a	2270.1	2270.1	160s	0.0%	2270.1	2270.1	958s	0.0%
		b	2615.2	2615.2	4809s	0.0%	2411.8	2740.1	10800s	11.9%
		c	2830.5	2830.5	166s	0.0%	2792.1	2830.5	10800s	1.3%
		d	2402.5	2402.5	384s	0.0%	2381.8	2416.0	10800s	1.4%
		e	2767.4	2767.4	22s	0.0%	2704.4	2777.7	10800s	2.6%
20	7	a	2970.4	2970.4	5048s	0.0%	2710.9	3092.7	10800s	12.3%
		b	3306.7	3369.7	10800s	1.8%	2757.0	3844.3	10800s	28.2%
		c	3527.8	3533.4	10800s	0.1%	2990.8	3838.3	10800s	22.0%
		d	2940.5	3012.4	10800s	2.3%	2675.0	3206.4	10800s	16.5%
		e	3534.7	3534.7	1926s	0.0%	2981.8	4313.9	10800s	30.8%
25	9	a	3702.9	3965.4	10800s	6.6%	3241.1	4512.6	10800s	28.8%
		b	4102.8	4379.2	10800s	6.3%	3428.7	5177.8	10800s	33.7%
		c	4339.3	4523.0	10800s	4.0%	3524.3	5207.9	10800s	32.3%
		d	3643.8	3780.9	10800s	3.6%	3237.5	4167.5	10800s	22.3%
		e	4230.9	4360.2	10800s	2.9%	3619.6	5764.1	10800s	37.2%
30	10	a	4005.5	4378.9	10800s	8.5%	3468.9	7927.2	10800s	56.2%
		b	4777.6	5114.0	10800s	6.5%	3869.3	7720.9	10800s	49.8%
		c	4711.3	4968.9	10800s	5.1%	3710.1	8391.2	10800s	55.7%
		d	4057.3	4340.6	10800s	6.5%	3506.2	7388.0	10800s	52.5%
		e	4808.1	5174.6	10800s	7.0%	3909.2	7844.2	10800s	50.1%

Tabela 1: Comparação dos resultados do algoritmo branch-and-price com a solução do modelo de fluxos pelo CPLEX usando até 4 threads

simultaneamente em até 4 threads, soluciona apenas 6 das 25 instâncias testadas, além de apresentar gaps de dualidade de até 56.2%. Considerando apenas aquelas instâncias resolvidas tanto pelo algoritmo branch-and-price quanto pelo modelo de fluxos, verifica-se que o tempo deste segundo é no mínimo 2.5 vezes maior que o alcançado pelo algoritmo branch-and-price.

O modelo (1)-(8) apresenta como inconveniente a simetria, pois indexa uma mesma rota para $|K|$ veículos, dificultando a convergência do algoritmo branch-and-price. No entanto, quando a frota de veículos é homogênea esta simetria se agrava, uma vez que a capacidade e os custos para atravessar os arcos são os mesmos para qualquer veículo. O próximo experimento foi proposto para avaliar a influência da simetria na obtenção de soluções pelo algoritmo branch-and-price. As mesmas instâncias serão solucionadas considerando diferentes níveis de heterogeneidade da frota, que serão mensurados pela quantidade de veículos iguais na frota.

A Tabela 2 apresenta os resultados do algoritmo branch-and-price para as instâncias com 20 requisições e 7 veículos, sendo que nas colunas 2 e 3 estão respectivamente o tempo de execução do algoritmo e o gap de dualidade obtido dentro de um limite de tempo de três horas para uma frota heterogênea, onde todos os veículos são distintos. As mesmas entradas se repetem nas colunas 4 e 5 para uma frota parcialmente homogênea, onde metade dos veículos são iguais e nas colunas 6 e 7 para uma frota de veículos homogênea.

instância	Frota Heterogênea		Parcial. Homogênea		Frota Homogênea	
	tempo	GAP	tempo	GAP	tempo	GAP
20a-k7	5048s	0.0%	10800s	0.2%	10800s	4.2%
20b-k7	10800s	1.8%	10800s	3.1%	10800s	3.8%
20c-k7	10800s	0.1%	10800s	2.8%	10800s	3.1%
20d-k7	10800s	2.3%	10800s	3.4%	10800s	6.5%
20e-k7	1926s	0.0%	10800s	0.1%	10800s	1.7%

Tabela 2: Influência da heterogeneidade da frota na avaliação de soluções

Os resultados da Tabela 2 mostram que o desempenho do algoritmo branch-and-price é melhor para frotas heterogêneas, devido à simetria inerente ao modelo de geração de colunas (1)-(8). Quando todos os veículos da frota são distintos, o algoritmo encontra soluções ótimas para duas instâncias de 20 requisições e alcança soluções com gaps de dualidade médio de 1.4% para aquelas instâncias não resolvidas no limite de tempo. No caso em que metade dos veículos da frota é igual, nenhuma solução ótima foi encontrada dentro do limite de tempo e o gap médio alcançado foi de 1.9%, enquanto para frotas homogêneas este valor foi de 3.6%.

5. Considerações Finais

A utilização de armazéns de *Cross-Docking* tem se tornado prática comum entre empresas de médio de grande porte na distribuição de mercadorias. O PRVCD é um problema que surge da integração de um Problema de Roteamento de Veículos com o escalonamento das mercadorias no *Cross-Docking*. Entre os trabalhos sobre o PRVCD, nenhum considera frotas heterogêneas de veículos na distribuição. Este trabalho tem o propósito de apresentar um algoritmo exato para solucionar o PRVCD com frotas heterogêneas.

Uma formulação matemática baseada em geração de colunas é apresentada bem como um algoritmo branch-and-price para solucioná-la. Experimentos foram conduzidos para avaliar o desempenho do algoritmo implementado e os resultados computacionais comprovaram sua eficiência na obtenção de soluções ótimas. Quando não foi possível avaliar soluções ótimas dentro de um limite de tempo, os algoritmo encontra soluções com gaps de dualidade reduzidos, principalmente quando comparados à solução de uma formulação de fluxos para o mesmo problema.

Entre as etapas futuras do trabalho está a avaliação de relaxações para a solução do sub-problema de Caminho Mínimo Elementar com Restrições de Recursos na etapa de geração de colunas. O uso de tais relaxações, como as *q-routes*, pode ser compensado com a inserção de planos de corte na formulação, levando à implementação de um algoritmo branch-and-cut-and-price para o problema.

Referências

- M. Uday Apte e S. Viswanathan**, Effective cross docking for improving distribution efficiencies, *International Journal of Logistics: Research and Applications*, 3:291–302, 2000.
- N. Boysen**, Truck scheduling at zero-inventory cross docking terminals, *Computers and Operations Research*, 37:32–41, 2010.
- D. Feillet, P. Dejax, M. Gendreau, e C. Gueguen**, An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems, *Networks*, 44:216–229, 2004.

- Gary Forger**, Ups starts world's premiere cross-docking operation, *Modern Materials Handling*, páginas 36–38, 1995.
- Kevin R. Gue**, Cross-docking: Just-in-time for distribution, Teaching Notes-Naval Post-graduate School – Monterey, CA, May 2001.
- G.F. King e C.F Mast**, Excess travel: Causes, extent and consequences, *Transportation Research Record*, páginas 126–134, 1997.
- Young Hae Lee, Jung Woo Jung, e Kyong Min Lee**, Vehicle routing scheduling for cross-docking in the supply chain, *Computers and Industrial Engineering*, 51(2):247–256, 2006, ISSN 0360-8352.
- F. A. Santos, A. S. Cunha, e G. R. Mateus**, Modelos de otimização para o problema de roteamento de veículos com cross-docking, *Simpósio Brasileiro de Pesquisa Operacional - SBPO*, 2010.
- F. A. Santos, G. R. Mateus, e A. S. Cunha**, A novel column generation algorithm for the vehicle routing problem with cross-docking, *International Network Optimization Conference - INOC*, 2011a, aceito para publicação.
- F. A. Santos, G. R. Mateus, e A. S. Cunha**, A branch-and-price algorithm for a vehicle routing problem with cross-docking, *Latin-American Algorithms, Graphs Optimization Symposium - LAGOS*, 2011b.
- Min Wen, Jesper Larsen, Jean Clausen, Jean-Francois Cordeau, e Gilbert Laporte**, Vehicle routing with cross-docking, *Journal of Operational Research Society*, 2008, ISSN 0160-5682, doi: 10.1057/jors.2008.108.
- Clyde E. Witt**, Crossdocking: Concepts demand choice, *Material Handling Engineering*, 53:44–49, 1998.
- Wooyeon Yu e Pius J. Egbelu**, Scheduling of inbound and outbound trucks in cross docking systems with temporary storage, *European Journal of Operational Research*, 184(1):377–396, 2008.