

Planeamento de Rotas num Sistema de Recolha de Desperdícios de Madeira

Cláudio Manuel Martins Alves * José Manuel Valério de Carvalho *

* Departamento de Produção e Sistemas, Universidade do Minho
{claudio,vc}@dps.uminho.pt

Abstract

In this paper, we analyze a new vehicle routing problem, the Prize Collecting Vehicle Routing Problem with Service Restrictions (PCVRPsr), which arises in wood-waste collection. It is a problem with a homogeneous fleet and a unique depot, in which the visit to some clients is not compulsory, but conditioned by the total needs of waste. We propose a formulation for this problem, which comes from a three-index vehicle flow formulation for the Vehicle Routing Problem.

To optimize the plan, we explore methods based on decomposition. We analyze, in particular, the Dantzig-Wolfe decomposition applied to this formulation, and a branch-and-price scheme to find integer solutions.

An algorithm was developed to obtain the computational results discussed at the end of the paper. In the branch-and-price tree, we applied a method for deriving lower bounds for the bin-packing problem, based on dual feasible functions, that makes the process more efficient.

Resumo

Neste artigo, analisa-se um novo problema de planeamento de rotas, o *Prize Collecting Vehicle Routing Problem with service restrictions* (PCVRPsr), sugerido por um caso de recolha de desperdícios de madeira. É um problema onde a frota é homogénea, o depósito único e em que a visita a alguns clientes não é obrigatória, mas condicionada pelas necessidades totais de desperdícios. Propõe-se uma formulação para este problema que deriva de um modelo de fluxo de três índices para o problema de planeamento de rotas de veículos.

Para a otimização do plano de rotas, exploraram-se métodos de decomposição. Analisa-se, em particular, a aplicação do método de decomposição de Dantzig-Wolfe à formulação proposta e, para a obtenção de soluções inteiras, o método de partição e geração de colunas (*branch-and-price*).

Foi desenvolvido um algoritmo com o qual se obtiveram os resultados computacionais que analisamos na parte final do artigo. No algoritmo de pesquisa em árvore, foi aplicado um método de determinação de limites inferiores para o problema de empacotamento, baseado em funções duais válidas, que torna o processo mais eficiente.

Keywords: Routing, Transportation, Integer Programming

Title: Vehicle Routing in a Wood Waste Pickup System

1 Introdução

O sistema de recolha em estudo é constituído por um conjunto de unidades distribuídas (fábricas de serração, superfícies comerciais, produtores de mobiliário) que geram desperdícios de madeira como fruto da sua actividade produtiva e por um fabricante de aglomerados que se encarrega de recolher com os seus próprios meios aquilo que para ele representa uma matéria prima barata. Nesse contexto, um dos problemas que enfrenta o fabricante consiste em determinar um plano económico de recolhas atendendo às condições específicas do seu sector.

O fabricante tem um duplo objectivo. Por um lado, recolher os desperdícios necessários à sua laboração. Por outro, assumir plenamente o seu papel de agente ecológico, garantindo o serviço aos clientes cujos níveis de desperdícios acumulados estejam acima de um determinado patamar. Distinguem-se assim duas classes de clientes: os que têm que ser obrigatoriamente visitados e aqueles que só o são se as necessidades periódicas do fabricante não puderem ser satisfeitas com os primeiros. Desta forma, um cliente com um nível de desperdícios baixo pode não ser visitado. Do ponto de vista económico, o planeamento visa a minimização dos custos variáveis de operação dos veículos, considerados proporcionais à distância total percorrida (como acontece frequentemente na prática, assume-se que as matrizes de distâncias e tempos de trânsito obedecem ao princípio de desigualdade triangular).

Para tal, o fabricante dispõe de uma frota virtualmente ilimitada de veículos de igual capacidade e de um depósito único. As descargas são exclusivamente efectuadas nesse depósito, onde se iniciam e terminam todas as rotas. A duração de cada rota é naturalmente limitada à duração de um turno de trabalho. Os clientes, as unidades cujos desperdícios de madeira são recolhidos, não impõem qualquer restrição à operação dos veículos. Um veículo pode visitar um cliente a qualquer momento do dia, procedendo de imediato à recolha, com a única garantia do levantamento da totalidade dos desperdícios disponíveis. Dessa forma, no horizonte de um planeamento, um cliente é visitado por apenas um veículo.

No problema clássico de planeamento de rotas de veículos, designado na literatura anglo-saxónica por *Vehicle Routing Problem* (VRP), todos os clientes têm de ser visitados. No *Prize Collecting Travelling Salesman Problem* introduzido por Balas [3], essa condição é relaxada: um cliente pode não ser visitado incorrendo-se por isso numa penalização contabilizada directamente no custo total da solução. No problema em estudo, não visitar um cliente com um nível baixo de desperdícios implica uma economia imediata dos custos de deslocação até esse cliente. Contudo, ao abdicar dos seus desperdícios, incorre-se numa penalização indirecta igual ao custo da visita a outro cliente do mesmo tipo, visita que passa a ser necessária em virtude das necessidades do fabricante. Os prémios que define o PCTSP correspondem agora aos níveis de desperdícios de alguns clientes. Por essas razões, designaremos o problema por *Prize Collecting Vehicle Routing Problem with service restrictions* (PCVRPsr). As restrições de serviço referem-se à imposição de serviço aos clientes com contentores cheios.

Na Secção 2, apresentamos um modelo genérico de fluxo de três índices para o VRP.

O modelo de fluxo de três índices para o PCVRPsr, introduzido na Secção 3, deriva desse modelo genérico. Na Secção 4, aplica-se uma decomposição de Dantzig-Wolfe a esse modelo, e são definidos o problema mestre e o subproblema. Na Secção 5, apresenta-se a estratégia usada no método de partição e geração de colunas. Na Secção 6, são mostrados resultados computacionais relativos a instâncias de teste com diferentes características, e, finalmente, na Secção 7, são retiradas algumas conclusões sobre a aplicação desta abordagem a este novo problema.

2 Modelo de Fluxo de Três Índices para o VRP

O VRP pode ser definido numa rede cujos elementos representam informação no espaço e no tempo ou simplesmente no espaço. Neste trabalho, recorreremos a uma rede caracterizada por nodos que representam localidades a qualquer instante de tempo e por arcos que traduzem exclusivamente movimentos no espaço.

Define-se o grafo $G = (V, A)$, em que V representa o conjunto dos nodos e A o conjunto dos arcos orientados. Seja $N = \{1, \dots, n\}$ o conjunto dos clientes onde existe um contentor susceptível de ser recolhido. O depósito é representado pelos nodos o e d , que o distinguem enquanto primeira origem e destino final dos veículos. Um arco (i, j) não terá nunca por origem o nodo d ou por destino o nodo o . Os conjuntos V e A ficam completamente definidos por $V = N \cup \{o, d\}$ e $A = \{N \cup \{o\} \times N \cup \{d\}\}$.

A cada cliente i está associada uma quantidade p_i de desperdícios de madeira disponíveis para recolha. Assume-se que $p_o = p_d = 0$. Entre cada par de nodos i e j existe um tempo de trânsito t_{ij} no qual se inclui o tempo de serviço ao cliente i . Temos ainda que $t_{ij} = t_{ji}, \forall i, j \in V$, e $t_{ij} \leq t_{ih} + t_{hj}, \forall i, j, h \in V$.

Um arco (i, j) , com $i, j \in N$, define uma sequência válida de recolhas nos clientes i e j . Os arcos (o, j) e (j, d) representam respectivamente o primeiro e último movimento de qualquer veículo, precedido ou seguido por uma recolha no cliente j . Quanto ao arco (o, d) , na prática, ele corresponde a não sair do depósito. Uma rota consiste finalmente numa sequência de recolhas atribuídas a um veículo da frota. Seja K o conjunto desses veículos. Todos os veículos têm igual capacidade Q e tempo máximo de trânsito T .

A cada arco (i, j) está associado um custo unitário c_{ij} , igual à menor distância, directa ou indirecta, entre os nodos i e j . De notar que $c_{od} = 0$. Tal como para a matriz de tempos de trânsito, a matriz de distâncias é simétrica e verifica o princípio de desigualdade triangular. Os custos fixos relativo à utilização de veículos (necessariamente igual para todos eles), se existirem, são incluídos nos custos $c_{oj}, j \in N$, dado que cada veículo sai uma única vez do depósito. Esses custos são particularmente importantes nos casos em que se pretende minimizar o número de veículos usados.

Sem perda de generalidade, assume-se que todos os parâmetros são inteiros não-negativos.

O modelo de fluxo de três índices [10] que adoptaremos para o VRP, e a partir do qual formularemos um modelo para o PCVRPsr, deriva de uma formulação proposta em [6, 7, 8] para o VRP com janelas temporais (VRPTW), para o qual são impostos intervalos de tempo admissíveis para as visitas. As variáveis $X_{ij}^k, (i, j) \in A, k \in K$, são variáveis de fluxo que tomam o valor 1 se o arco (i, j) pertencer à rota do veículo k e 0 caso contrário. As variáveis

L_i^k e $T_i^k, i \in V, k \in K$, representam, após a passagem pelo nodo i , a carga do veículo k e a duração do seu trajecto desde que partiu do depósito, respectivamente. Os veículos saem vazios do depósito, i.e. $L_o^k = 0, \forall k \in K$.

$$\min. \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} X_{ij}^k \quad (1)$$

sujeito a

$$\sum_{k \in K} \sum_{j \in N \cup \{d\}} X_{ij}^k = 1, \quad \forall i \in N \quad (2)$$

$$\sum_{j \in N \cup \{d\}} X_{oj}^k = 1, \quad \forall k \in K \quad (3)$$

$$\sum_{i \in N \cup \{o\}} X_{ij}^k - \sum_{i \in N \cup \{d\}} X_{ji}^k = 0, \quad \forall k \in K, \forall j \in V \setminus \{o, d\} \quad (4)$$

$$\sum_{i \in N \cup \{o\}} X_{id}^k = 1, \quad \forall k \in K \quad (5)$$

$$X_{ij}^k (T_i^k + t_{ij} - T_j^k) \leq 0, \quad \forall k \in K, \forall (i, j) \in A \quad (6)$$

$$0 \leq T_i^k \leq T, \quad \forall k \in K, \forall i \in V \quad (7)$$

$$X_{ij}^k (L_i^k + p_i - L_j^k) \leq 0, \quad \forall k \in K, \forall (i, j) \in A \quad (8)$$

$$0 \leq L_i^k \leq Q, \quad \forall k \in K, \forall i \in V \quad (9)$$

$$X_{ij}^k \geq 0, \quad \forall k \in K, \forall (i, j) \in A. \quad (10)$$

Claramente, sempre que o problema (1)-(10) for possível, existirá uma solução óptima inteira [7]. Contudo, essa formulação é não linear e define um espaço de soluções não convexo. A formulação (1)-(10) pode ser linearizada substituindo as restrições (6) e (8) pelas equações (11) e (12) e adicionando a condição de integralidade (13):

$$T_i^k + t_{ij} - T_j^k \leq (1 - X_{ij}^k)M, \quad \forall k \in K, \forall (i, j) \in A \quad (11)$$

$$L_i^k + p_i - L_j^k \leq (1 - X_{ij}^k)Q, \quad \forall k \in K, \forall (i, j) \in A \quad (12)$$

$$X_{ij}^k \text{ binário}, \quad \forall k \in K, \forall (i, j) \in A. \quad (13)$$

M representa um valor suficientemente grande para que as restrições sejam redundantes para $X_{ij}^k = 0$.

A função (1) representa o custo total das rotas. A restrição (2) garante a visita de todos os clientes por um único veículo. Um veículo k não utilizado corresponde a um veículo que sai do depósito-origem o e regressa directamente ao seu depósito-destino d , i.e. $X_{od}^k = 1$. O trajecto de um veículo k é descrito pelas restrições de conservação de fluxo (3)-(5). As restrições (6) e (7) garantem que a duração máxima de cada rota não seja excedida enquanto as restrições (8) e (9) garantem que a capacidade dos veículos não seja também excedida. Existem formas alternativas de definir as restrições de capacidade e duração que conduzem a formulações com um número bastante inferior de variáveis e restrições. Contudo, essa aparente ineficiência faz-se sem prejuízo do método de resolução adoptado e que apresentaremos mais adiante.

3 Modelo de Fluxo de Três Índices para o PCVRPsr

O PCVRPsr consiste numa generalização do VRP com restrições adicionais relativas à recolha de uma quantidade mínima de desperdícios de madeira .

O fabricante de aglomerados necessita de ns unidades de desperdícios de madeira. Por seu lado, os clientes não admitem guardar mais de nc unidades desse material. Acima das nc unidades, um cliente tem de ser visitado; abaixo, poderá ou não sê-lo, consoante as necessidades do fabricante. Seja N_1 o conjunto de clientes do primeiro tipo (os clientes cheios) e N_2 o conjunto de clientes do segundo tipo (os clientes semi-cheios). Essas condições, que designamos por restrições de serviço, distinguem o nosso problema do que seria um PCVRP clássico, no qual não é feita nenhuma distinção entre clientes. Se fizermos $nc = \min_{i \in N} p_i$, obtém-se um VRP clássico. Com $nc = \sum_{i \in N} p_i$, o problema reduz-se a um PCVRP clássico.

Se existirem clientes cheios, apenas uma parte das necessidades do fabricante terá de ser satisfeita através dos clientes semi-cheios. Seja ns_v essa quantidade. Temos que $ns_v = ns - \sum_{i \in N_1} p_i$. Claramente, se $ns \leq \sum_{i \in N_1} p_i$, o problema reduz-se a um VRP clássico.

Baseados no modelo (1)-(13), podemos formular o PCVRPsr da seguinte forma:

$$\min. \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} X_{ij}^k \quad (14)$$

sujeito a

$$\sum_{k \in K} \sum_{j \in N \cup \{d\}} X_{ij}^k = 1, \quad \forall i \in N_1 \quad (15)$$

$$\sum_{k \in K} \sum_{j \in N \cup \{d\}} X_{ij}^k \leq 1, \quad \forall i \in N_2 \quad (16)$$

$$\sum_{k \in K} \sum_{i \in N_2} p_i \sum_{j \in N \cup \{d\}} X_{ij}^k \geq ns_v \quad (17)$$

$$(3)-(5), (7) \text{ e } (9)-(13)$$

A restrição (2) desdobra-se agora nas restrições (15) e (16). A primeira impõe a visita de todos os clientes cheios enquanto a restrição (16) relaxa essa condição no caso dos clientes semi-cheios. Em (17) garante-se a recolha das ns_v unidades através dos clientes semi-cheios.

A dimensão do problema condiciona fortemente a sua resolução. Como referimos no final da Secção anterior, é possível reduzir a dimensão do modelo reformulando as restrições de capacidade e duração através de uma única inequação por veículo, como é feito usualmente. Evitava-se assim o recurso às variáveis T_i^k e L_i^k . A economia é importante, mas a dimensão do problema continua considerável. A formulação proposta nesta Secção define $|K| \times ((N+1)^2 - N)$ variáveis de fluxo, $|K| \times N$ variáveis T_i^k e um igual número de variáveis de carga L_i^k . O maior número de restrições do modelo está concentrado em (11) e (12). Em cada conjunto, o número de restrições ascende a $|K| \times ((N+1)^2 - N)$.

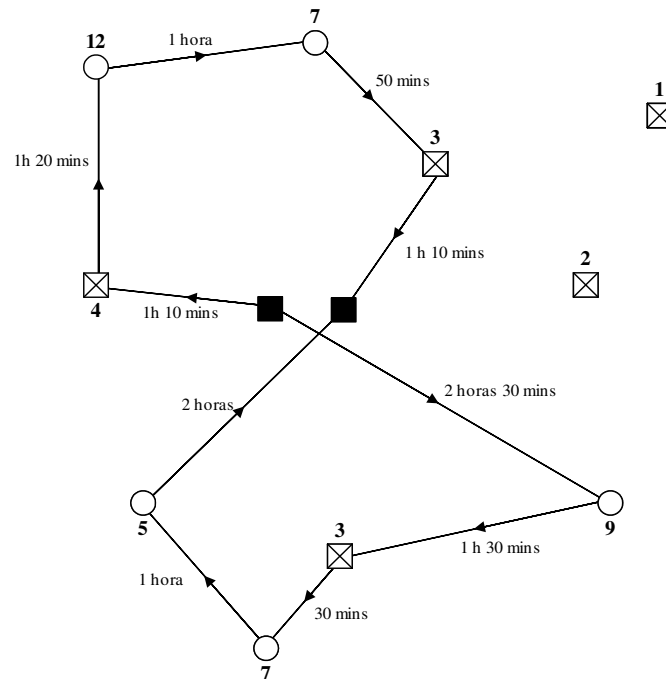


Figura 1: Solução válida para a instância do Exemplo 1.

Exemplo 1 Considere uma instância do PCVRPsr com 10 clientes, 5 veículos, cujo grafo completo é parcialmente definido na Figura 1. As necessidades periódicas de recolha são de 50 toneladas. A capacidade dos veículos é de 30 toneladas e a duração das rotas é limitada a 1 dia. O nível limite de stock é de 5 unidades. Qualquer cliente com um nível de stock acima dessas 5 unidades é considerado cliente cheio, e tem consequentemente de ser visitado. Na Figura 1, o depósito, enquanto origem e destino final, é representado pelos quadrados a preto. Os clientes cheios são representados pelos círculos. O valor perto dos símbolos indica a carga respectiva de cada cliente. A figura apresenta uma solução válida para essa instância na qual dois veículos são usados para servir os clientes com 12,7,5,7 e 9 unidades de desperdícios. As 10 unidades em falta são recolhidas em três outros clientes.

Para essa instância, a formulação desta Secção tem 655 variáveis e 1161 restrições. Com as simplificações sugeridas para as restrições de capacidade e duração, o número de variáveis passaria para 555, enquanto o número de restrições cairia para 75. ■

É possível derivar diversos casos especiais notáveis a partir do PCVRPsr. Se fixarmos o limite nc de stock tolerado no menor valor de stock possível, relaxarmos as restrições de capacidade e duração e limitarmos o problema a um único veículo, obtém-se um problema de caixeiro viajante. O problema de empacotamento resulta da fixação de nc no seu valor mínimo, da relaxação da restrição de duração e das seguintes alterações dos custos: $c_{ij}^k = 1$, $k \in K$, $i = o(k)$, $j \in N$ e $c_{ij}^k = 0$ caso contrário. Ambos os problemas são NP-difíceis [11], o que faz do PCVRPsr um problema também NP-difícil.

O problema que consiste em determinar se existe uma solução válida para determinada instância do PCVRPsr, quando o número de veículos é livre, é um problema de fácil resolução. Basta associar um veículo a cada cliente cheio e completar, eventualmente, as necessidades,

associando sequencialmente um veículo a cada cliente semi-cheio. No pior dos casos, este procedimento requer n operações. Se o número de veículos for fixo, o mesmo problema é NP-completo. Por essa razão, muitos dos algoritmos desenvolvidos no âmbito dos problemas de planeamento de rotas assumem que o número de veículos é livre. Esta conclusão, válida também para o VRPTW que é um caso especial do PCVRP_{sr}, é um corolário do seguinte resultado obtido no caso de um único veículo sem limite de capacidade [14]:

Teorema 3.1 *O problema de decisão que consiste em saber se determinada instância do problema do caixeiro viajante com janelas temporais tem solução é NP-completo, mesmo quando os tempos de trânsito são simétricos.*

4 Decomposição de Dantzig-Wolfe

Aplicando o método de decomposição de Dantzig-Wolfe [5] ao modelo da Secção anterior, é possível deixar as restrições de capacidade e duração das rotas para um subproblema de caminho mais curto com restrições adicionais, definindo-se um problema mestre de cobertura de conjuntos.

4.1 Problema Mestre

Dada a formulação da Secção anterior para o PCRVP_{sr}, a decomposição que propomos define o problema mestre a partir das restrições (10)-(17) com a respectiva função objectivo (14) inalterada e corresponde a um problema de partição de conjuntos com restrições adicionais. O subproblema é composto pelas restrições (3)-(5), (11), (7), (12) e (9)-(13) e consiste originalmente em $|K|$ problemas idênticos de caminho mais curto sem ciclos com restrições de capacidade e duração, dos quais apenas um é efectivamente considerado.

O poliedro definido pelas restrições do subproblema é limitado. Assim, as variáveis de fluxo X_{ij}^k podem ser expressas como uma combinação convexa e não negativa dos pontos extremos desse poliedro, que são caminhos no grafo $G = (V, A)$. Se X_{ijp} representar o valor do fluxo no arco (i, j) de um ponto extremo $p \in \Omega$ do poliedro do subproblema (com Ω a representar o conjunto de todos os seus pontos extremos), teremos que:

$$X_{ij}^k = \sum_{p \in \Omega} \lambda_p^k X_{ijp}, \quad \forall k \in K, \forall (i, j) \in A$$

com as seguintes restrições:

$$\sum_{p \in \Omega} \lambda_p^k = 1, \quad \forall k \in K$$

$$\lambda_p^k \geq 0 \quad \forall k \in K, \forall p \in \Omega$$

É possível associar a cada caminho $p \in \Omega$ um conjunto de informação. Assim, o número de vezes que um caminho p visita um cliente i e que designaremos por a_{ip} é igual a $\sum_{j \in N \cup \{d\}} X_{ijp}$.

Numa solução válida, $a_{ip} \in \{0, 1\}$. O custo total c_p de um caminho p é de $\sum_{(i,j) \in A} c_{ij} X_{ijp}$

unidades. Quanto ao volume l_p de desperdícios provenientes de clientes semi-cheios que são

recolhidos ao longo de um percurso p , temos que $l_p = \sum_{i \in N_2} p_i \sum_{j \in N \cup \{d\}} X_{ijp}$. Fazendo ainda $\lambda_p = \sum_{k \in K} \lambda_p^k$, podemos rescrever o problema mestre da seguinte forma:

$$\min. \sum_{p \in \Omega} \lambda_p c_p \quad (18)$$

sujeito a

$$\sum_{p \in \Omega} \lambda_p a_{ip} = 1, \quad \forall i \in N_1 \quad (19)$$

$$\sum_{p \in \Omega} \lambda_p a_{ip} \leq 1, \quad \forall i \in N_2 \quad (20)$$

$$\sum_{p \in \Omega} \lambda_p l_p \geq ns.v, \quad (21)$$

$$\lambda_p \text{ binário}, \quad \forall p \in \Omega \quad (22)$$

Da formulação de fluxos em arcos original passamos a uma formulação equivalente com fluxos em caminhos. Devido à estratégia de partição escolhida (Subsecção 5.1), adicionaram-se ao modelo duas variáveis X_c e X_d iguais respectivamente ao custo total da solução e ao número de veículos efectivamente usados.

$$\sum_{p \in \Omega} \lambda_p c_p = X_c \quad (23)$$

$$\sum_{p \in \Omega \setminus \{(o,d)\}} \lambda_p = X_d \quad (24)$$

Para reduzir problemas de estabilidade numérica associadas ao modelo de partição de conjuntos com restrições adicionais (18)-(24) que foram detectados na resolução de problemas com estrutura semelhante [6], e pelo facto do subproblema poder, como veremos na próxima Subsecção, gerar colunas com ciclos ($a_{ip} \geq 1$), optou-se por um modelo de cobertura de conjuntos relaxando a restrição (19):

$$\sum_{p \in \Omega} \lambda_p a_{ip} \geq 1, \quad \forall i \in N_1 \quad (25)$$

Mostra-se facilmente que o conjunto de soluções do problema de cobertura de conjuntos é idêntico ao do problema de partição de conjuntos, devido à propriedade de desigualdade triangular [1].

A Figura 2 ilustra a estrutura do problema mestre. O problema é inicializado com um número restrito de colunas e resolvido através do método de geração de colunas. Quando a procura dos clientes cheios é suficiente para cobrir as necessidades do fabricante, apenas são geradas rotas de ida e volta entre esses clientes e o depósito (os restantes clientes são obviamente excluídos do modelo). Caso contrário, adiciona-se uma rota por cada cliente semi-cheio na ordem crescente dos seus índices até que seja garantida a satisfação total das necessidades. A variável artificial *Art* foi incluída para garantir a validade dos problemas em qualquer nodo da árvore de pesquisa. O seu custo M é muito elevado e os seus coeficientes nas restrições (25) e (20) igual à unidade.

	X_c	X_d	λ_1	...	$\lambda_{ \Omega }$		
Clientes cheios	0	0	a_{11}	...	$a_{1 \Omega }$	\geq	1
	0	0	\geq	1
	0	0	$A_{ N_1 1}$...	$A_{ N_1 \Omega }$	\geq	1
Clt. semi-cheios	0	0	$a_{ N_1 +1,1}$...	$a_{ N_1 +1, \Omega }$	\leq	1
	0	0	\leq	1
	0	0	$a_{ N_1 + N_2 ,1}$...	$a_{ N_1 + N_2 , \Omega }$	\leq	1
Necessidades	0	0	l_1	...	l_n	\geq	ns_v
N.º de veículos	0	-1	1	...	1	=	0
Custo total	-1	0	c_1	...	c_n	=	0
	0	0	c_1	...	c_n		

Figura 2: Quadro Simplex do problema de cobertura de conjuntos com restrições adicionais.

4.2 Subproblema

O Problema de Caminho Mais Curto Sem Ciclos e com Restrições de Capacidade e Duração (PCMSCRCRD) que resulta de uma simplificação da formulação (3)-(5), (11), (7), (12) e (9)-(13), escolhida para o subproblema, e que não possui a propriedade da integralidade, é NP-difícil, e, em geral, de difícil resolução. Por esse motivo, recorreremos a uma formulação relaxada do PCMSCRCRD na qual se elimina a condição de elementaridade imposta aos caminhos. O problema continua a ser um problema NP-difícil mas para o qual são conhecidos algoritmos pseudo-polinomiais. Na solução inteira óptima do problema mestre, onde o *overcovering* dos clientes nunca ocorrerá, as colunas associadas aos caminhos com ciclos serão todas nulas.

Pretende-se encontrar caminhos cujas colunas associadas no problema mestre tenham custo reduzido negativo. Designem-se por π_i, π_l, π_c e π_d as variáveis duais do problema mestre restrito associadas às restrições (25) e (20) para cada cliente $i \in N$ e às restrições (21), (23) e (24), respectivamente. O custo reduzido c'_p de um caminho $p \in \Omega$ é dado por:

$$c'_p = c_p - \sum_{i \in N_1 \cup N_2} \pi_i a_{ip} - \pi_l l_p - \pi_c c_p - \pi_d$$

É possível, por substituição dos parâmetros a_{ip}, l_p e c_p e alguns arranjos de termos, chegar à seguinte equação para o custo reduzido do caminho p expresso em ordem aos custos dos arcos de G :

$$c'_p = \sum_{i \in N_1} \sum_{j \in N \cup \{d\}} (c_{ij} - \pi_i - \pi_c c_{ij}) X_{ijp} + \sum_{i \in N_2} \sum_{j \in N \cup \{d\}} (c_{ij} - \pi_i - \pi_c c_{ij} - \pi_l p_i) X_{ijp} + \sum_{j \in N} (c_{oj} - \pi_c c_{oj} - \pi_d) X_{ojp} + (c_{od} - \pi_c c_{od} - \pi_d) X_{odp}$$

Os custos dos arcos do subproblema, os custos marginais, baseiam-se nos custos originais e são calculados da seguinte forma:

$$\begin{aligned} c'_{ij} &= c_{ij} - \pi_i - \pi_c c_{ij} && \text{se } i \in N_1 \\ c'_{ij} &= c_{ij} - \pi_i - \pi_c c_{ij} - \pi_l p_i && \text{se } i \in N_2 \\ c'_{ij} &= c_{ij} - \pi_c c_{ij} - \pi_d && \text{se } i = o \text{ e } j \in N \\ c'_{ij} &= c_{ij} - \pi_c c_{ij} - \pi_d && \text{se } i = o \text{ e } j = d \end{aligned}$$

O subproblema consiste assim numa relaxação do seguinte PCMCSCRCO:

$$\min. \sum_{(i,j) \in A} c'_{ij} X_{ij} \quad (26)$$

sujeito a

$$\sum_{j \in N \cup \{d\}} X_{oj} = 1, \quad (27)$$

$$\sum_{i \in N \cup \{o\}} X_{ij} - \sum_{i \in N \cup \{d\}} X_{ji} = 0, \quad \forall j \in V \setminus \{o, d\} \quad (28)$$

$$\sum_{i \in N \cup \{o\}} X_{id} = 1, \quad (29)$$

$$T_i + t_{ij} - T_j \leq (1 - X_{ij})M, \quad \forall (i, j) \in A \quad (30)$$

$$0 \leq T_i \leq T, \quad \forall i \in V \quad (31)$$

$$L_i + p_i - L_j \leq (1 - X_{ij})Q, \quad \forall (i, j) \in A \quad (32)$$

$$p_i \leq L_i \leq Q, \quad \forall i \in V \quad (33)$$

$$X_{ij} \geq 0, \quad \forall (i, j) \in A \quad (34)$$

$$X_{ij} \text{ binário}, \quad \forall (i, j) \in A. \quad (35)$$

$$(36)$$

O subproblema é resolvido através de um algoritmo de programação dinâmica de complexidade pseudo-polinomial, também usado em [6] para o problema muito semelhante de caminho mais curto com restrições de capacidade e janelas temporais.

5 Pesquisa de Soluções Inteiras: Método de Partição e Geração de Colunas

O modelo do PCVRPsr não possui a propriedade da integralidade. A solução óptima inteira pode ser obtida através do método de partição e geração de colunas (*branch-and-price*) que consiste numa combinação do método de partição e avaliação sucessivas (*branch-and-bound*) com o método de geração de colunas [4]. Em cada nodo da árvore de pesquisa, o limite inferior dado pela solução óptima da relaxação linear do PCVRPsr é calculada através do método de geração de colunas.

5.1 Estratégias de Partição

Se as partições baseadas nas variáveis de decisão do problema mestre podem resultar na regeneração das colunas correspondentes, no caso do problema em estudo, e, de modo geral, no caso dos problemas de planeamento de rotas formulados nos termos da decomposição de Dantzig-Wolfe, mostra-se que essa partição não é sequer praticável [2]. No lugar de uma partição nos caminhos, efectuou-se uma partição nas variáveis originais X_{ij} que correspondem a fluxos em arcos. Cada ramo da árvore de pesquisa resulta da adição das restrições $X_{ij}=1$ e $X_{ij}=0$.

Como essas variáveis não fazem parte da formulação do problema mestre, é necessário derivar um procedimento equivalente na formulação de cobertura de conjuntos que apresentámos na Secção anterior.

As modificações são executadas na matriz de custos do problema. Para garantir que o arco (i, j) não faz parte de nenhuma solução óptima ($X_{ij}=0$), fixa-se um valor elevado ao respectivo custo c_{ij} e eliminam-se do problema mestre as colunas que correspondam a caminhos que passam por esse arco.

Para garantir a presença do arco (i, j) na solução óptima do problema, penalizam-se todos os arcos que saem do nodo i e todos aqueles que terminam em j , excepto o arco (i, j) , fixando em valores elevados os custos correspondentes. O custo das colunas que incluem esses arcos é actualizado ao mesmo tempo que o valor da função objectivo do problema mestre. Dessa forma, espera-se que uma rota seja forçada a visitar o cliente j imediatamente após o cliente i .

Essa estratégia só faz sentido quando i e j são clientes cheios. Para esses casos, um fluxo fraccionário num caminho que atravessa os clientes i e j implica a existência de outros caminhos que os visitam com fluxos igualmente fraccionários e passando por arcos diferentes de (i, j) (se não existisse fluxo em quaisquer outros arcos incidentes em i e j , sem contar com o arco (i, j) , o fluxo X_{ij} seria igual a 1 e não chegaria a ser escolhido para a partição). Com as penalizações introduzidas, a solução óptima deixa efectivamente de o ser porque tem arcos com custos elevados. A cobertura dos clientes i e j só pode ser garantida pelo arco (i, j) , o que obriga assim o seu fluxo a assumir o valor $X_{ij}=1$. O mesmo acontece quando apenas um dos clientes é cliente cheio.

Nos casos em que i e j são clientes semi-cheios, o esquema que apresentámos não garante por si só que o fluxo no arco (i, j) seja mesmo unitário. Possivelmente, poderá mesmo ser nulo se o cliente i não pertencer à solução óptima do novo problema (os custos de todos os arcos incidentes em j têm valores elevados à excepção do arco (i, j)). Lembramos que os clientes semi-cheios são sujeitos a restrições do tipo $\sum_{p \in \Omega} \lambda_p a_{ip} \leq 1$, podendo, na relaxação do problema inteiro, atravessá-los um fluxo total fraccionário. Se na solução óptima do problema de partida todos os arcos que saem de i e todos os arcos que terminam em j tiverem fluxo nulo, as transformações feitas na matriz de custos não terão qualquer efeito na solução óptima do novo problema. Os clientes i e j terão de passar a ser explicitamente considerados como clientes cheios, o que implica substituir as restrições correspondentes do tipo " \leq " por restrições de cobertura usuais do tipo " \geq ". Genericamente, esse procedimento foi adoptado sempre que a partição envolvia um cliente semi-cheio.

Uma partição do tipo $X_{ij}=1$ resulta numa agregação dos clientes i e j . A criação de um novo problema onde $X_{ij}=1$ só pode ser feita se a soma das duas cargas não exceder a capacidade dos veículos e o tempo de trânsito entre eles também não for superior ao limite de duração imposto. Caso uma dessas restrições fosse violada e fosse, mesmo assim, criado o problema onde $X_{ij}=1$, o problema resultaria numa solução de custo muito elevado, previsivelmente não atractiva. É necessário ainda tomar em consideração as agregações efectuadas a níveis superiores na árvore de pesquisa. Assim, se um cliente l , por exemplo, tiver já sido agregado ao cliente i , a partição $X_{ij}=1$ só será possível se a soma das cargas de i , j e l for inferior à capacidade dos veículos. O mesmo se aplica no caso das durações. Esses testes são inseridos na regra de selecção, inspirada em [6].

As partições baseiam-se ainda nas variáveis de custo total X_c e do número de veículos usados X_d , introduzidas especialmente para esse efeito na formulação do problema mestre.

5.2 Limite Inferior do Problema de Empacotamento para Detecção de Nós Impossíveis

Em alguns nodos da árvore de pesquisa, a partição efectuada a partir de X_d , na qual se impõe um limite superior ao número de veículos, pode conduzir a um problema inválido. A detecção oportuna dessa situação permite evitar operações de processamento inúteis. Para tal, bastaria comparar um limite inferior ao número de veículos necessários com o limite superior imposto pela partição.

Esse limite inferior é dado pelo limite inferior para a solução do problema de empacotamento das cargas dos clientes cheios (ou que se vão tornando cheios em virtude da regra de partição) nos veículos de capacidade limitada, e pode ser calculado com base numa função dual válida apresentada em [9]. Considere, sem perda de generalidade, um problema de empacotamento normalizado, onde a capacidade dos contentores é unitária e as cargas distribuídas entre 0 e 1. Uma função $u : [0,1] \rightarrow [0,1]$ é uma função dual válida se para qualquer conjunto S de números reais não negativos x se verificar:

$$\sum_{x \in S} x \leq 1 \Rightarrow \sum_{x \in S} u(x) \leq 1$$

É válida a seguinte proposição enunciada em Fekete e Schepers (1998):

Proposição 5.1 *Se $I = (x_1, \dots, x_n)$ for uma instância do problema de empacotamento e u uma função dual válida, então o limite inferior para a instância modificada do problema de empacotamento $u(I) = (u(x_1), \dots, u(x_n))$ é também um limite inferior para I .*

O cálculo de um limite inferior para o problema de empacotamento definido a partir das cargas dos clientes cheios passa pelo cálculo de limites inferiores para o problema de empacotamento transformado através de uma função dual válida. Um limite inferior trivial para um problema de empacotamento $I = (x_1, \dots, x_n)$ normalizado é dado por:

$$L_1(I) = \left\lceil \sum_{i=1}^n x_i \right\rceil$$

As funções duais válidas apresentadas por Fekete e Schepers (1998) são técnicas de arredondamento das cargas cujo objectivo consiste em aumentar o mais possível o valor das cargas para assim obter limites inferiores de melhor qualidade. Consideramos apenas uma delas:

$$u^{(k)} : [0, 1] \rightarrow [0, 1]$$

$$x \mapsto \begin{cases} x, & \text{se } x(k+1) \in \mathbb{Z} \\ \lfloor (k+1)x \rfloor \frac{1}{k}, & \text{caso contrário} \end{cases}$$

Calculam-se limites para diferentes valores de $k \in \mathbb{Z}$. No final, o limite inferior para o número de veículos corresponderá ao maior desses valores. Outros limites inferiores para o problema de empacotamento podem ser encontrados em [13].

Os limites inferiores são calculados antes que seja tomada uma decisão de partição baseada em X_d e sempre que tenha havido nova agregação de cargas. Os cálculos tomam evidentemente em conta essa agregação, o que implica a gestão de estruturas de dados adicionais que permitam o acesso a uma imagem corrigida do grafo do problema. Por exemplo, dois clientes que tenham sido agregados passam a contar por um único nodo com uma carga associada igual à soma das cargas dos clientes. É necessário registar essa informação em cada nodo da árvore de pesquisa por forma a garantir que futuras agregações sejam feitas tendo em atenção agregações previamente realizadas.

Inicialmente, apenas se conhece um número restrito de clientes que irão ser visitados (os clientes cheios). O limite inferior para o número de veículos necessários só pode ser calculado com base nas suas cargas. Consequentemente, para baixos níveis de profundidade na árvore de pesquisa, não são de esperar limites de muito boa qualidade. Contudo, quanto maior for a profundidade, maior será o número de clientes semi-cheios que passam para o grupo dos clientes cheios, e melhor será o limite inferior. Por outro lado, se as restrições de capacidade dos veículos forem menos restritivas do que as restrições de duração das rotas, dificilmente o limite chegará a ser efectivo. O Exemplo 2 ilustra o cálculo deste limite inferior.

Exemplo 2 Considere uma instância particular do PCVRPsr com 5 clientes e na qual o nível limite de stock foi fixado no menor valor aceitável (VRP). A capacidade dos veículos é de 10 unidades de carga. Para ilustrar como se pode obter limites inferiores de melhor qualidade do que o limite inferior trivial $L_1 = \left\lceil \sum_{i=1}^n x_i \right\rceil$, em que x_i é a carga normalizada do cliente i , vamos usar a função dual válida com $k=3$. As cargas individuais de cada cliente são, respectivamente, iguais a 5, 5, 4, 3 e 3. O valor do limite inferior para o número de veículos calculado no primeiro nodo da árvore de pesquisa é de 2 veículos (o mesmo que L_1). O processo seguido para o seu cálculo foi sintetizado na tabela seguinte:

Cliente i	p_i	$x_i = p_i/10$	$x_i (k+1)$	$u^{(3)}(x_i)$
1	5	0.5	2	0.5
2	5	0.5	2	0.5
3	4	0.4	1.6	1/3
4	3	0.3	1.2	1/3
5	3	0.3	1.2	1/3
				2

De facto, é possível empacotar os 2 itens com peso 5 num veículo e os itens com pesos 4, 3 e 3, respectivamente, noutro veículo. Essa solução pode não corresponder ao empacotamento que se verifica na solução óptima do PCVRPsr. Constitui, contudo, um limite inferior ao número de veículos necessários.

Admite-se que a solução óptima da relaxação linear do problema mestre tem um número fraccionário de veículos compreendido entre 2 e 3 e que, no ramo onde foi inserida a restrição $X_d \leq 2$, é escolhido o arco (2, 3) para a próxima partição. Para um dos problemas novos que deveria ser criado, aquele onde $X_{23}=1$, a configuração das cargas passaria a ser a que se representa na tabela seguinte:

Cliente i	p_i
1	5
2 e 3	9
4	3
5	3

O limite L_1 continua a ser igual a 2. Recorrendo à função dual válida, o limite inferior que se obtém nesse caso passa a ser aproximadamente 2.17 veículos, conforme se ilustra no quadro seguinte:

Cliente i	p_i	$x_i = p_i/10$	$x_i (k+1)$	$u^{(3)}(x_i)$
1	5	0.5	2	0.5
2 e 3	9	0.9	3.6	1
4	3	0.3	1.2	1/3
5	3	0.3	1.2	1/3
				2.17

Claramente esse valor entra em conflito com os dois veículos máximos impostos pela anterior restrição de partição. Identificada essa situação, apenas o problema onde $X_{23}=0$ é considerado.

6 Resultados Computacionais

As instâncias de teste foram geradas aleatoriamente, segundo distribuições uniformes e intervalos de valores predefinidos. As instâncias geradas podem ser divididas em três grupos. Num primeiro grupo, a sombreado escuro na Tabela 1, estão as instâncias que representam problemas estritamente de *prize collecting* (PCVRP). Não possuem nenhum cliente cheio pelo que as visitas aos clientes visa apenas cobrir as necessidades periódicas. A sombreado claro na mesma tabela encontram-se as instâncias que possuem pelo menos um cliente semi-cheio (e pelo menos um cliente cheio), e cujas necessidades não são totalmente cobertas pelas cargas dos clientes cheios. Nessa situação, um conjunto de clientes semi-cheios terá mesmo de ser visitado. No último grupo, temos as instâncias cujas necessidades são cobertas pelas cargas dos clientes cheios. Essas instâncias definem VRPs de dimensão igual ou inferior a N .

Apresentam-se apenas as instâncias com 15 e 25 clientes nas quais as capacidades dos veículos e o limite imposto à duração das rotas são factores efectivamente restritivos e para as quais foi possível obter uma solução óptima em tempos razoáveis. Para instâncias de PCVRPs com mais de 50 clientes, geradas segundo o mesmo processo aleatório, observa-se uma degradação considerável dos tempos de resolução. A duração máxima imposta às rotas, que define uma espécie de janela temporal dentro da qual se deve efectuar uma visita num cliente, é geralmente larga e dificulta a resolução do subproblema de programação dinâmica. A Tabela 1 resume as principais características das 27 instâncias utilizadas.

O primeiro problema mestre restrito é definido segundo uma adaptação da estratégia "um cliente – um veículo" ao caso dos PCVRPs. Ao fim de cada resolução do subproblema, são inseridas, se existirem, mais que uma coluna atractiva. O problema mestre é resolvido através

Tabela 1: Instâncias de Teste.

	N	Número de clientes semi-cheios	Carga dos clientes cheios	Carga dos clientes semi-cheios	Necessidades	Necessidades Residuais	Capacidade dos veículos
1	15	0	53	0	45	0	9
2	15	15	0	139	44	44	23
3	15	15	0	53	42	42	9
4	15	0	231	0	67	0	41
5	15	0	54	0	38	0	8
6	15	0	30	0	20	0	2
7	15	15	0	54	35	35	8
8	15	0	35	0	7	0	5
9	15	1	60	2	27	0	8
10	15	13	8	33	23	15	7
11	15	15	0	216	101	101	41
12	15	0	45	0	25	0	8
13	15	8	28	24	35	7	8
14	15	0	177	0	99	0	34
15	15	6	29	12	34	5	6
16	15	5	187	45	101	0	38
17	15	15	0	248	78	78	44
18	25	0	61	0	28	0	5
19	25	25	0	391	386	386	45
20	25	21	36	105	49	13	14
21	25	0	60	0	21	0	4
22	25	0	57	0	46	0	4
23	25	25	0	499	104	104	47
24	25	0	379	0	46	0	32
25	25	8	58	16	66	8	7
26	25	14	91	94	139	48	17
27	25	19	156	286	260	104	41

do Simplex primal. O algoritmo de programação dinâmica não gera nenhum caminho que tenha ciclos de ordem inferior ou igual a dois. A estratégia de pesquisa usada foi do tipo "primeiro em profundidade".

Nas tabelas de resultados adiante apresentadas, regista-se na primeira coluna o índice da instância, conforme a ordem definida na Tabela 1. A segunda coluna identifica a dimensão da instância, caracterizada pelo número de clientes. A terceira coluna representa o número de veículos efectivamente usados na solução óptima inteira. A quarta coluna regista o número de colunas geradas no processo de optimização da relaxação linear do problema mestre. As três colunas seguintes indicam os valores obtidos no método de partição e geração de colunas (número total de colunas geradas ao longo da árvore de pesquisa, número de nodos e profundidade máxima alcançada). Os tempos de execução são registados nas três colunas seguintes, distinguindo-se o tempo gasto na resolução do problema mestre da raiz da árvore de pesquisa e o tempo gasto nos restantes nodos e indicando o tempo total. As duas últimas colunas indicam o valor da solução óptima fraccionária da relaxação linear do problema inteiro original e a sua solução óptima inteira. Os resultados foram obtidos num computador com processador Intel Pentium II, frequência de relógio de 450Mhz e 64 Mbytes de memória RAM, usando o CPLEX 6.0 [12].

Na Tabela 2, apresentam-se os resultados obtidos para as instâncias que possuem pelo menos um cliente cheio e um cliente semi-cheio, e cujas necessidades residuais são não nulas. A Tabela 3 apresenta uma síntese idêntica de resultados para as instâncias da Tabela 1 que são VRPs clássicos.

Para as instâncias representadas na Tabela 2, observa-se um tempo médio de resolução da relaxação linear da ordem dos 42% do tempo total usado na resolução do problema inteiro. Para as instâncias da Tabela 3, esse valor sobe para os 61.6%. Esta situação sugere a necessidade do recurso a heurísticas para a inicialização do problema mestre mais sofisticadas do que aquela que foi usada.

A diferença entre as duas percentagens de tempo é ainda apoiada pela diferença sensível que existe entre os intervalos de integralidade para os dois casos. Para as instâncias da Tabela 2, o intervalo médio de integralidade é de 5.4% (sendo o maior de 17,2% para a instância n.º 2), enquanto que para as instâncias listadas na Tabela 3 esse valor é de apenas 0.9% (o maior fica-se pelos 3,3% para a instância 8). A percentagem de tempo gasto no método de partição e geração de colunas, para o segundo caso, decresce em consequência da diminuição do intervalo de integralidade a pesquisar. Uma conclusão imediata consiste em reconhecer a perda de qualidade do limite inferior fornecido pelo modelo de cobertura com restrições adicionais quando aplicado a instâncias que representam PCVRPs que tenham pelo menos um cliente semi-cheio a visitar. O aumento do intervalo médio de integralidade justifica em grande parte os 58% do tempo total de execução gastos no método de partição e geração de colunas na resolução das instâncias da Tabela 2.

Quando as instâncias do PCVRPs têm um rácio entre as necessidades residuais e a carga total dos clientes semi-cheios muito elevado (próximo dos 100%), o intervalo de integralidade tende a diminuir. A relação entre as cargas dos clientes semi-cheios e as necessidades residuais influenciou em alguns casos o desempenho do algoritmo. Constatou-se de facto que os tempos de resolução pioravam em algumas instâncias onde as proporções de clientes semi-cheios com cargas baixas e necessidades residuais elevadas eram importantes.

A transformação das instâncias da Tabela 2 em instâncias de VRPs confirmam as conclusões que tirámos há pouco em relação ao aumento do intervalo médio de integralidade. Com essa transformação, o intervalo médio decresceu significativamente, passando dos 5.4% para os 1.2%.

Na Tabela 4, listam-se os resultados obtidos para as instâncias de VRPs obtidas a partir das instâncias da Tabela 2. Comparando os tempos de resolução das instâncias do PCVRPs (Tabela 2) com os tempos de resolução das suas correspondentes transformações em VRPs (Tabela 3), observa-se que, em 6 casos apenas, as instâncias do primeiro grupo levaram mais tempo a serem resolvidas. Em 5 desses casos, o intervalo de integralidade para as instâncias do PCVRPs era maior do que aquele obtido para as instâncias do VRP. Contudo, outros casos há em que apesar do intervalo de integralidade ser maior, o tempo de resolução dos PCVRPs é menor. É o caso, por exemplo, da instância n.º2. Para os restantes 8 casos, os tempos de resolução das instâncias do PCVRPs é menor.

Novamente, verificamos que a maior parcela do esforço computacional, medida em função do tempo de processamento, é despendida em fases distintas da resolução. Para as instâncias do PCVRPs o tempo continua a concentrar-se nas iterações do método de partição e geração de colunas, enquanto que para as instâncias do VRP o tempo é maioritariamente gasto na

Tabela 2: Resultados computacionais para as instâncias do PCVRPsr com necessidades residuais.

	N	Veíc.	cols RL	cols MPGC	nodos MPGC	prof.	tempo RL	tempo MPGC	tempo total	solução RL	solução inteira
2	15	2	6	2	25	11	51.41	5.72	57.13	659.60	797.00
3	15	6	13	30	125	19	1.71	2.80	4.51	4777.33	5194.00
7	15	5	14	29	199	22	0.33	5.60	5.93	3107.50	3586.00
10	15	4	19	11	2	1	0.72	0.49	1.21	2259.92	2266.00
11	15	3	5	0	8	5	0.06	0.22	0.28	1834.81	1861.00
13	15	5	31	2	150	20	0.11	2.63	2.74	3446.33	3606.00
15	15	6	28	9	89	18	0.43	1.54	1.97	4472.17	4689.00
17	15	2	17	0	67	18	86.4	70.36	156.76	1346.90	1487.00
19	25	22	4	0	0	0	0.04	0.00	0.04	16900.00	16900.00
20	25	5	8	0	5	3	0.11	0.22	0.33	3295.29	3353.00
23	25	3	19	1	47	14	0.71	2.04	2.75	1098.70	1266.00
25	25	11	65	0	43	16	6.21	1.10	7.31	8842.00	8888.00
26	25	9	44	25	29	10	0.28	0.82	1.10	7352.00	7397.00
27	25	7	44	31	312	24	5.16	17.25	22.41	5496.15	5562.00

Tabela 3: Resultados computacionais para as instâncias do PCVRPsr sem necessidades residuais (VRP_s).

	N	Veíc.	cols RL	cols MPGC	nodos MPGC	prof.	tempo RL	tempo MPGC	tempo total	solução RL	solução inteira
1	15	6	56	8	2	1	32.35	1.98	34.33	6138.00	6139.00
4	15	7	30	0	0	0	0.11	0.00	0.11	6801.00	6801.00
5	15	7	46	0	12	7	0.61	0.38	0.99	6254.00	6352.00
6	15	15	0	0	0	0	0.05	0.00	0.05	11870.0	11870.00
8	15	8	28	0	11	5	0.06	0.16	0.22	5804.50	6005.00
9	15	9	23	0	2	1	0.06	0.05	0.11	7202.50	7248.00
12	15	6	58	0	2	1	16.70	0.05	16.75	6029.83	6107.00
14	15	6	64	7	40	16	3.57	3.36	6.93	4551.00	4663.00
16	15	6	23	0	2	1	34.27	17.52	51.79	4703.00	4763.00
18	25	13	73	2	83	15	0.11	2.31	2.42	10697.00	10791.00
21	25	20	20	0	2	1	0.06	0.05	0.11	17879.00	17911.00
22	25	16	54	4	74	12	0.05	1.76	1.81	14606.00	14644.00
24	25	13	83	12	57	14	24.55	4.94	29.49	10692.00	10752.00

resolução da relaxação linear. De facto, em quase todos os casos, a resolução da relaxação linear foi mais rápida para as instâncias do PCVRPsr do que para as instâncias do VRP. A única excepção é a instância 5.

O efeito do limite inferior do problema de empacotamento foi avaliado para os casos mais favoráveis do PCVRPsr que correspondem a instâncias do VRP. O que se observa para as restantes instâncias é uma degradação progressiva da eficácia em função do número de clientes semi-cheios. Nesses casos, o limite inferior só consegue actuar a partir de determinados níveis de profundidade, quando o conjunto de clientes cheios começa a ganhar maior preponderância. Os resultados que apresentamos nas Tabelas 5 e 6 correspondem à aplicação do nosso algoritmo às instâncias iniciais da Tabela 1 transformadas em instâncias do VRP. Parte da Tabela 5 foi já apresentada na Tabela 4.

São nove as instâncias que viram o seu tempo total de resolução diminuir. Em média, se considerarmos todas as instâncias, a economia no tempo total de resolução é de 4.9% (para a instância 26 o tempo é reduzido em quase 50%; se retirarmos essa instância a média passa para 3.15%). Se tomarmos em consideração apenas as instâncias em que o limite inferior é útil, a economia sobe para os 14.8%. Esse valor é significativo, comprovando a vantagem da utilização de limites inferiores para o problema de empacotamento no problema de VRP.

7 Conclusões

Neste artigo, explorámos um novo problema de planeamento de rotas, o *Prize Collecting Vehicle Routing Problem with service restrictions* (PCVRPsr). O problema aparece como uma generalização de um *Vehicle Routing Problem* (VRP) clássico, no qual se relaxa em parte a condição de visita obrigatória aos clientes. Distiguem-se assim dois grupos de clientes: os clientes cheios com níveis de desperdícios acima de determinado limite e os clientes semi-cheios com níveis abaixo desse limite. A visita aos clientes semi-cheios só é efectuada se as necessidades do fabricante não puderem ser satisfeitas com base nos desperdícios dos clientes cheios.

Formulou-se um modelo matemático de fluxo de três índices, ao qual se aplicou a decomposição de Dantzig-Wolfe, que resultou num problema mestre de partição de conjuntos com restrições adicionais, relaxado posteriormente num modelo de cobertura de conjuntos, e num subproblema de caminho mais curto sem ciclos e com restrições de capacidade e duração, que acabamos por resolver relaxando a condição de elementaridade dos caminhos. Para a pesquisa de soluções inteiras, recorreu-se ao método de partição e geração de colunas. Calcularam-se também limites inferiores para o problema de empacotamento, baseados em funções duais válidas, para a detecção antecipada de nodos impossíveis na árvore de pesquisa, o que resultou em economias de tempo apreciáveis no problema de VRP.

Verificou-se um aumento sensível do intervalo de integralidade fornecido pelo modelo de cobertura de conjuntos aplicado a instâncias do PCVRPsr com pelo menos um cliente semi-cheio a visitar. Para as instâncias do VRP, que são casos especiais do PCVRPsr, esse intervalo era sensivelmente menor. Futuras abordagens deverão procurar formulações mais fortes que resultem em intervalos de integralidade reduzidos. Quanto à relaxação linear inicial, os tempos de resolução foram em geral bem menores no caso do PCVRPsr mas poderiam ser certamente reduzidos recorrendo-se a heurísticas para obtenção do primeiro problema mestre restrito.

Tabela 4: Resultados computacionais para as instâncias do PCVRPsr transformadas em VRPs.

	N	Veíc.	cols RL	cols MPGC	nodos MPGC	prof.	tempo RL	tempo MPGC	tempo total	solução RL	solução inteira
2	15	7	46	11	61	17	63.27	16.64	79.91	4478.50	4751.00
3	15	6	50	3	2	1	4.73	1.37	6.10	5613.00	5661.00
7	15	7	39	0	0	0	0.77	0.00	0.77	5893.00	5893.00
10	15	6	59	4	70	13	1.20	1.93	3.13	5276.67	5401.00
11	15	11	10	0	5	3	0.06	0.05	0.11	9584.50	9782.00
13	15	8	40	4	11	6	0.11	0.49	0.60	5569.50	5574.00
15	15	7	30	0	10	6	0.22	0.33	0.55	5766.67	5819.00
17	15	6	52	23	31	12	198.94	14.94	213.88	5889.00	5973.00
19	25	22	6	0	0	0	0.05	0.00	0.05	18088.00	18088.00
20	25	21	11	0	0	0	0.12	0.00	0.12	17821.00	17821.00
23	25	11	78	97	139	16	1.76	9.23	10.99	9629.50	9890.00
25	25	11	125	14	34	10	10.82	4.89	15.71	9659.50	9680.00
26	25	13	67	25	318	21	0.28	17.41	17.69	11194.00	11289.00
27	25	12	92	36	73	15	6.70	5.72	12.42	10713.20	10755.00

Tabela 5: Resultados computacionais (sem recurso ao limite inferior do problema de empacotamento) para as instâncias da Tabela 1 transformadas em instâncias do VRP.

	N	Veíc.	cols RL	Cols MPGC	Nodos MPGC	prof.	tempo RL	tempo MPGC	tempo total	solução RL	solução inteira
1	15	6	56	8	2	1	32.35	1.98	34.33	6138.00	6139.00
2	15	7	46	11	61	17	63.27	16.64	79.91	4478.50	4751.00
3	15	6	50	3	2	1	4.73	1.37	6.10	5613.00	5661.00
4	15	7	30	0	0	0	0.11	0.00	0.11	6801.00	6801.00
5	15	7	46	0	12	7	0.61	0.38	0.99	6254.00	6352.00
6	15	15	0	0	0	0	0.05	0.00	0.05	11870.00	11870.00
7	15	7	39	0	0	0	0.77	0.00	0.77	5893.00	5893.00
8	15	8	28	0	11	5	0.06	0.16	0.22	5804.50	6005.00
9	15	10	31	0	7	4	0.06	0.11	0.17	7654.83	7695.00
10	15	6	59	4	70	13	1.20	1.93	3.13	5276.67	5401.00
11	15	11	10	0	5	3	0.06	0.05	0.11	9584.50	9782.00
12	15	6	58	0	2	1	16.70	0.05	16.75	6029.83	6107.00
13	15	8	40	4	11	6	0.11	0.49	0.60	5569.50	5574.00
14	15	6	64	7	40	16	3.57	3.36	6.93	4551.00	4663.00
15	15	7	30	0	10	6	0.22	0.33	0.55	5766.67	5819.00
16	15	7	58	0	2	1	78.93	0.00	78.93	5924.50	5960.00
17	15	6	52	23	31	12	198.94	14.94	213.88	5889.00	5973.00
18	25	13	73	2	83	15	0.11	2.31	2.42	10697.00	10791.00
19	25	22	6	0	0	0	0.05	0.00	0.05	18088.00	18088.00
20	25	21	11	0	0	0	0.12	0.00	0.12	17821.00	17821.00
21	25	20	20	0	2	1	0.06	0.05	0.11	17879.00	17911.00
22	25	16	54	4	74	12	0.05	1.76	1.81	14606.00	14644.00
23	25	11	78	97	139	16	1.76	9.23	10.99	9629.50	9890.00
24	25	13	83	12	57	14	24.55	4.94	29.49	10692.00	10752.00
25	25	11	125	14	34	10	10.82	4.89	15.71	9659.50	9680.00
26	25	13	67	25	318	21	0.27	17.41	17.68	11194.00	11289.00
27	25	12	92	36	73	15	6.70	5.72	12.42	10713.20	10755.00

Tabela 6: Resultados computacionais (com recurso ao limite inferior do problema de empacotamento) para as instâncias da Tabela 1 transformadas em instâncias do VRP.

	N	Veíc.	cols RL	cols MPGC	nodos MPGC	prof.	Tempo RL	tempo MPGC	tempo total	solução RL	solução inteira
1	15	6	56	8	2	1	32.35	1.98	34.33	6138.00	6139.00
2	15	7	46	11	61	17	63.27	16.64	79.91	4478.50	4751.00
3	15	6	50	3	2	1	4.73	1.37	6.10	5613.00	5661.00
4	15	7	30	0	0	0	0.11	0.00	0.11	6801.00	6801.00
5	15	7	46	0	12	7	0.61	0.38	0.99	6254.00	6352.00
6	15	15	0	0	0	0	0.05	0.00	0.05	11870.00	11870.00
7	15	7	39	0	0	0	0.77	0.00	0.77	5893.00	5893.00
8	15	8	28	0	10	5	0.05	0.17	0.22	5804.50	6005.00
9	15	10	31	0	7	4	0.06	0.11	0.17	7654.83	7695.00
10	15	6	59	4	70	13	1.20	1.93	3.13	5276.67	5401.00
11	15	11	10	0	5	3	0.06	0.05	0.11	9584.50	9782.00
12	15	6	58	0	1	1	16.70	0.00	16.70	6029.83	6107.00
13	15	8	40	1	10	6	0.11	0.39	0.50	5569.50	5574.00
14	15	6	64	9	35	16	3.62	3.16	6.78	4551.00	4663.00
15	15	7	30	0	9	6	0.22	0.33	0.55	5766.67	5819.00
16	15	7	58	0	1	1	78.82	0.00	78.82	5924.50	5960.00
17	15	6	52	23	31	12	198.94	14.94	213.88	5889.00	5973.00
18	25	13	73	0	82	15	0.11	2.20	2.31	10697.00	10791.00
19	25	22	6	0	0	0	0.05	0.00	0.05	18088.00	18088.00
20	25	21	11	0	0	0	0.06	0.00	0.06	17821.00	17821.00
21	25	20	20	0	2	1	0.06	0.05	0.11	17879.00	17911.00
22	25	16	54	4	74	12	0.05	1.76	1.81	14606.00	14644.00
23	25	11	78	97	139	16	1.76	9.23	10.99	9629.50	9890.00
24	25	13	83	16	56	14	24.77	2.53	27.3	10692.00	10752.00
25	25	11	125	14	34	10	10.77	4.83	15.6	9659.50	9680.00
26	25	13	67	14	313	21	0.28	8.40	8.68	11194.00	11289.00
27	25	12	92	36	73	15	6.7	5.72	12.42	10713.20	10755.00

Agradecimentos

Gostaríamos de agradecer ao revisor as sugestões e comentários que ajudaram a tornar mais claro este artigo.

8 Referências

- [1] Alves, C.M., *Planeamento de rotas de veículos em sistemas de recolha/distribuição*, Dissertação de Mestrado em Engenharia Industrial, Especialidade de Logística e Distribuição, Universidade do Minho, 126 pp., Dezembro de 2000.
- [2] Ahuja, R., T. Magnanti e J. Orlin, *Network Flows: Theory, Algorithms and Applications*, Prentice-Hall (1993).
- [3] Balas, E., *The Prize Collecting Traveling Salesman Problem*, *Networks* 19 (1989), pp. 621-636.
- [4] Barnhart, C., E. Johnson, G. Nemhauser, M. Savelsbergh e P. Vance, *Branch-and-price: Column Generation for Solving Huge Integer Programs*, *Operations Research* 46 (1998), pp.316-329.
- [5] Dantzig G. B. e P. Wolfe, *Decomposition Principle for Linear Programs*, *Operations Research* 8 (1960), pp. 101-111
- [6] Desrochers, M., J. Desrosiers e M. Solomon, *A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows*, *Operations Research* 40 (1992), pp. 342-354.
- [7] Desrosiers, J., F. Soumis e M. Desrochers, *Routing with Time Windows by Column Generation*, *Networks* 14 (1984), pp. 545-565
- [8] Desrosiers J., Y. Dumas, M. Solomon e F. Soumis, *Time Constrained Routing and Scheduling*, *Handbooks in OR & MS* 6 (1995), pp. 35-139.
- [9] Fekete, S. e J. Schepers, *New Classes of Lower Bounds for Bin-Packing Problems*, *Lecture Notes in Computer Science* 1412 (1998).
- [10] Fisher, M.L. e R. Jaikumar, *A Generalized Assignment Heuristic for Vehicle Routing*, *Networks* 11, (1981), pp. 109-124.
- [11] Garey, M. e D. Johnson, *Computers Intractability*, W.H. Freeman and Company (1979), San Francisco, USA.
- [12] Ilog, Inc., *Using the CPLEX Callable Library*, Version 6.0 (1998).
- [13] Martello, S. e P. Toth, *Knapsack Problems: Algorithms and Computer Implementations*, John Wiley and Sons (1990).
- [14] Savelsbergh, M., *Local Search in Routing Problems with Time Windows*. *Annals of Operations Research* 4 (1985), pp. 285-305.