

Formulação de programação inteira para um problema de consolidação de cargas com frete morto e custos por múltiplas entregas

Mateus C. Silva, Christian Elster, Rafael A. Melo

Instituto de Computação, Universidade Federal da Bahia

Salvador, BA, Brasil

`mateuscs@ufba.br, christian.elster@ufba.br, rafael.melo@ufba.br`

RESUMO

Neste artigo, consideramos um problema de consolidação de cargas no qual uma empresa remetente precisa entregar vários itens aos seus clientes em um único dia usando veículos terceirizados de múltiplos transportadores. Cada veículo possui informações de tipo, capacidade, carga mínima contratada, e tipos de itens que pode transportar. Um veículo pode visitar diferentes clientes se estiverem dentro de uma distância máxima permitida. Ademais, os preços dependem do tipo de veículo, do número de entregas (clientes diferentes visitados), do custo de transporte por unidade de peso dos itens consolidados, e do frete morto correspondente à disponibilidade não utilizada da carga mínima contratada dos veículos. Mostramos que mesmo uma variante restrita do problema é NP-difícil e propomos uma formulação de programação inteira. Experimentos preliminares utilizando instâncias sintéticas mostram que a distribuição de clientes em *clusters* favorece o melhor aproveitamento dos veículos e a encontrar boas soluções ao aumentar o número de itens.

PALAVRAS CHAVE. Consolidação de cargas. Frete morto. Programação inteira.

Tópicos (Logística e Transportes, Programação Matemática)

ABSTRACT

In this paper, we consider a freight consolidation problem arising in the context of road transport. In this problem, a shipper has to deliver multiple items to its customers in a single day using outsourced vehicles from multiple carriers. Besides, each vehicle has a type, a capacity, a minimum contracted load, and a set of item types it can carry. Moreover, a vehicle can visit different customers whenever they are within an allowed maximum distance. Additionally, the prices depend on the type of the vehicle, the number of deliveries (different visited customers), the per unit of weight transportation cost of the consolidated items, and the dead freight corresponding to the unused availability of the vehicles' minimum contracted load. First, we show that the problem is NP-hard. In addition, we propose a mixed-integer programming formulation for the problem. Preliminary experiments using synthetic instances show that the distribution of customers into clusters favors better utilization of the vehicles and finding good solutions by increasing the number of items.

KEYWORDS. Freight consolidation. Dead freight. Integer programming.

Paper topics (Logistics and Transportation, Mathematical Programming)

1. Introdução

Nas áreas de logística e transporte, problemas de consolidação de cargas (PCCs) apresentam desafios significativos para a alocação eficiente de recursos e otimização das operações. Muitas empresas empregam operadores logísticos terceirizados (3PLs) para o transporte de produtos. Em algumas situações, os 3PLs têm o desafio de consolidar pedidos de vários clientes para reduzir custos operacionais. Em outras, o remetente é responsável por garantir a consolidação dos pedidos dos seus clientes ao contratar um 3PL para o transporte. O transporte de produtos pode ser classificado nas categorias de longa ou curta distância. Os métodos de transporte de longa distância incluem o transporte rodoviário, marítimo e ferroviário [Christiansen et al., 2004]. Em casos, como no Brasil, onde predomina o transporte rodoviário [IBGE, 2014], o transporte de longa distância através das rodovias é amplamente utilizado. Neste trabalho, consideramos um PCC visando o transporte de longa distância. Neste problema, uma empresa remetente deve consolidar os pedidos de múltiplos clientes em uma frota heterôgenea de veículos de 3PLs. Os custos operacionais incluem um custo proporcional à carga transportada, penalidades de frete morto quando uma carga mínima contratada não é preenchida, e taxas associadas a múltiplas entregas realizadas por um veículo.

Hanbazazah et al. [2019] estudaram um PCC com remessas divisíveis, janelas de prazo de entrega e custos de transporte por partes. Devido à impraticabilidade de resolver o problema de forma exata usando programação inteira (PI), os autores adotaram uma estratégia de dividir o problema em três etapas: relaxar a carga do contêiner, decomposição temporal, e geração de cortes. Vieira e Munari [2021] apresentaram uma formulação de PI para um PCC com múltiplos períodos, frota heterôgenea, e custos por múltiplas entregas. O objetivo consiste em minimizar o custo de transporte e uma soma ponderada dos custos de itens não transportados.

Qin et al. [2014] introduziram o problema de containerização e consolidação de cargas (PCCC) cujo objetivo consiste em minimizar os custos totais de transporte de contêineres e entrega de produtos. O PCCC integra decisões de transporte de longa e curta distâncias. Os autores propuseram uma formulação de PI e um algoritmo memético. Considerando o mesmo problema, Melo e Ribeiro [2015] propuseram uma nova formulação por representativos com técnicas de agregação e quebra de simetria. Hajiaghahi-Keshteli et al. [2017] abordaram o PCCC com metaheurísticas populacionais hibridizadas. Tiwari et al. [2021] estenderam o PCCC com a introdução de custos relacionados a impostos sobre emissão de gás carbônico e apresentaram uma formulação de PI. Os autores realizaram uma análise de sensibilidade dos parâmetros do problema para verificar o grau de influência de cada um deles.

PCCs também compartilham semelhanças com problemas envolvendo centros de consolidação urbana (CCUs), os quais podem ser usados para consolidar serviços de entrega de cargas [Nataraj et al., 2019]. O PCC estudado neste trabalho compartilha ainda semelhanças com problemas de empacotamento [Jansen, 1999; Gendreau et al., 2004; Sadykov e Vanderbeck, 2013] e problemas de empacotamento com conflitos e caixas de tamanho variável [Ekici, 2022].

Neste trabalho, formalizamos um PCC com frete morto e custos por múltiplas entregas. Mostramos a NP-dificuldade do problema e propomos uma formulação de PI. O restante do trabalho está organizado como segue. A Seção 2 formaliza o problema estudado. A Seção 3 apresenta uma prova de NP-completude para a versão de decisão do problema. A Seção 4 descreve uma formulação de PI. A Seção 5 discorre sobre experimentos computacionais preliminares. A Seção 6 conclui o artigo com alguns comentários finais.

2. Formalização do problema

O PCC considerado neste artigo pode ser formalmente definido da seguinte forma. Existe um remetente (empresa que terceiriza sua entrega), um conjunto de itens de remessa I , um conjunto de clientes K , um conjunto de tipos de itens T , um conjunto de tipos de veículos $\mathcal{T}$, e um conjunto de veículos V terceirizados de diversas transportadoras, cada uma com suas próprias tarifas

e restrições de veículos. Cada item $i \in I$ tem um peso positivo w_i , um tipo $t_i \in T$, e um cliente associado $k_i \in K$, que define seu local de entrega. Além disso, denote o remetente por k_0 . Cada veículo $v \in V$ possui um tipo $\tau \in \mathcal{T}$, uma capacidade positiva C_v , uma carga mínima contratada não negativa $C_v^{\min}$, um preço não negativo p_v por unidade de peso por quilômetro rodado, um custo não negativo r_v por entrega adicional incorrido sempre que múltiplos clientes são visitados na mesma viagem (múltiplas entregas), um conjunto de tipos de itens permitidos T^v , um conjunto de clientes K^v que ele pode atender, determinado por suas localizações e pelos tipos de veículos que pode receber, e uma distância máxima não negativa entre clientes $maxd_v$ que podem ser visitados na mesma viagem. Cada veículo pode realizar no máximo uma viagem. Observe que uma viagem visitando um único cliente não implica custos de múltiplas entregas. Também, temos a distância não negativa $d_{kk'}$ entre cada par de clientes $k, k' \in K$ e a distância não negativa d_{k_0k} do remetente até cada cliente $k \in K$.

Denote por p_v^k o preço por unidade de peso para entrega dos itens do cliente $k \in K$ usando o veículo $v \in V$, ou seja, $p_v^k = p_v d_{k_0k}$. Além disso, seja V^i o conjunto de veículos que podem entregar o item $i \in I$, V^k o conjunto de veículos que podem atender o cliente $k \in K$, e I^v o conjunto de itens que podem ser entregues pelo veículo $v \in V$. Ademais, seja $\delta_{kk'}^v$ igual a um se os clientes $k, k' \in K$ puderem ser visitados na mesma viagem pelo veículo $v \in V$, ou seja, se $d_{kk'} \leq maxd_v$.

Considere um veículo consolidado $v \in V$, ou seja, um veículo com itens atribuídos a ele. Defina por λ_v sua carga total, α_v o cliente mais distante (mais caro) visitado por ele, e $p_v^{\alpha_v}$ o custo por unidade de peso para α_v . Um frete morto é pago sempre que v transporta menos que $C_v^{\min}$ unidades de peso, ou seja, $\lambda_v < C_v^{\min}$. O custo total de frete morto do veículo é dado por $\phi_v = p_v^{\alpha_v}(C_v^{\min} - \lambda_v)$.

O objetivo do problema consiste em obter uma atribuição viável de cada item a um veículo e clientes por ele atendidos de forma a minimizar o custo total, composto pelos custos de entrega dos itens, pelas tarifas de frete morto e pelos preços incorridos em múltiplas entregas, onde a reentrega dos itens pode ocorrer entre clientes próximos uns dos outros. Observe que todos os itens estão inicialmente localizados em um único local e devem ser distribuídos de forma eficiente para diversos destinos. Supõe-se que os veículos disponíveis sejam suficientes para a atribuição de todos os itens de remessa.

3. NP-dificuldade

A versão de decisão do PCC considerado consiste em decidir se existe uma atribuição viável dos itens aos veículos com custo menor ou igual a um valor de entrada $\mathcal{K}$. A seguir, mostramos que o problema é NP-completo por meio de uma redução da versão de decisão do problema de clique máximo (PCM), que pode ser formalizado da seguinte forma: Dado um grafo $G = (V', E)$ e um inteiro $\mathcal{K}'$, G contém uma clique com tamanho maior ou igual a $\mathcal{K}'$?

Proposição 1. *PCC é NP-completo.*

Demonstração. Primeiramente, argumentamos que $PCC \in NP$. Observe que dada uma solução para PCC, determinada por uma atribuição dos itens aos veículos, pode-se facilmente verificar em tempo polinomial se todas as restrições foram satisfeitas e se o custo é menor ou igual a $\mathcal{K}$.

Descrevemos agora uma transformação polinomial $PCM \propto PCC$. Dada uma instância para PCM, construímos uma instância para PCC da seguinte maneira. Para cada vértice $u \in V'$, crie um item $u \in I$ com peso $w_u = 1$, implicando que $|I| = |V'|$. Determine um único tipo para todos os itens e defina o conjunto de clientes K com $|I|$ clientes, um para cada item $u \in I$. Construa um conjunto de veículos $V = \{v_1, v_2\}$, com $C_{v_1} = C_{v_2} = |I| = |K| = |V'|$. Seja k_0 o remetente e defina $d_{k_0k_u} = 100$ para cada $u \in I$. Para cada aresta $uu' \in E$, defina $d_{k_u k_{u'}} = 10$ e para cada

anti-aresta $uu' \in \bar{E}$, $\bar{E} = \{uv \mid u, v \in V \text{ and } uv \notin E\}$, defina $d_{k_u k_{u'}} = 11$. Defina $maxd_1 = 10$ e $maxd_2 = 11$. Defina preços por unidade de peso $p_1 = 0.01$ e $p_2 = 0.01\mathcal{K}'$, implicando que $p_1^k = 1$ e $p_2^k = \mathcal{K}'$ para cada $k \in K$. Além disso, defina $\mathcal{K} = \mathcal{K}' + (|V'| - \mathcal{K}')\mathcal{K}'$. Todos os outros parâmetros de entrada são definidos como zero.

Resta mostrar que existe uma clique $K \in V'$ em G com pelo menos $\mathcal{K}'$ elementos se e somente se existe uma solução para PCC com custo menor ou igual a $\mathcal{K}$. Para isso, considere uma solução para a instância PCC criada e seja η o número de itens no veículo v_1 . Consequentemente, deve haver $|V'| - \eta$ itens no veículo v_2 e o custo dessa solução é $\eta + (|V'| - \eta)\mathcal{K}'$, uma vez que cada item possui peso unitário. Observe que cada aumento marginal de uma unidade em η diminui o custo da solução em $\mathcal{K}' - 1$, implicando que quanto maior for η , menor será o custo. Com isso em mente, considere uma clique com pelo menos $\mathcal{K}'$ elementos e deixe os itens correspondentes aos vértices da clique estarem no veículo v_1 e aqueles correspondentes aos vértices no complemento da clique estarem em v_2 . A solução é claramente viável para PCC e seu custo é no máximo $\mathcal{K}' + (|V'| - \mathcal{K}')\mathcal{K}' = \mathcal{K}$. Agora, considere uma solução para PCC com custo no máximo $\mathcal{K} = \mathcal{K}' + (|V'| - \mathcal{K}')\mathcal{K}'$. Como consequência, como foi observado anteriormente, segue-se que o número de itens em v_1 é $\eta \geq \mathcal{K}'$, e esses itens correspondem a vértices em V' que formam uma clique em G com pelo menos $\mathcal{K}'$ elementos. □

Observe que a Proposição 1 implica que a versão de otimização do PCC é NP-difícil mesmo quando há apenas dois veículos e não há custos relacionados a frete morto ou múltiplas entregas.

4. Formulação de programação inteira

A formulação de programação inteira (PI) mista apresentada nesta seção considera as seguintes variáveis de decisão. A variável binária x_{iv} é igual a um se o item $i \in I$ for atribuído ao veículo $v \in V$, zero caso contrário. A variável binária u_v é igual a um se o veículo $v \in V$ é usado, zero caso contrário. A variável binária y_{kv} é igual a um se o cliente $k \in K$ for visitado pelo veículo $v \in V$, zero caso contrário. A variável z_v define a quantidade de frete morto no veículo $v \in V$. A variável ϕ_v determina a tarifa de frete morto aplicada ao veículo $v \in V$. A variável b_v representa o número de entregas adicionais realizadas pelo veículo $v \in V$. O problema pode, ser formulado como:

$$\min \sum_{v \in V} \left(\sum_{i \in I^v} x_{iv} w_i p_v^{k_i} + \phi_v + b_v r_v \right) \quad (1)$$

$$\sum_{v \in V^i} x_{iv} = 1, \quad \forall i \in I, \quad (2)$$

$$\sum_{i \in I^v} w_i x_{iv} \leq C_v u_v, \quad \forall v \in V, \quad (3)$$

$$x_{iv} \leq y_{k_i v}, \quad \forall i \in I, v \in V^i, \quad (4)$$

$$y_{kv} + y_{k'v} \leq u_v, \quad \forall k, k' \in K, v \in V^k \cap V^{k'}, \text{ com } k \neq k' \text{ e } \delta_{kk'}^v = 0, \quad (5)$$

$$y_{kv} \leq u_v, \quad \forall k \in K, v \in V^k, \quad (6)$$

$$b_v \geq \sum_{k \in K^v} y_{kv} - 1, \quad \forall v \in V, \quad (7)$$

$$\sum_{i \in I^v} w_i x_{iv} + z_v \geq C_v^{\min} u_v, \quad \forall v \in V, \quad (8)$$

$$z_v \leq C_v^{\min} u_v, \quad \forall v \in V, \quad (9)$$

$$\phi_v \geq p_v^k z_v - p_v^k (1 - y_{kv}) C_v^{\min}, \quad \forall v \in V, k \in K^v, \quad (10)$$

$$x_{iv} \in \{0, 1\}, \quad \forall i \in I, v \in V^i, \quad (11)$$

$$u_v \in \{0, 1\}, \quad \forall v \in V, \quad (12)$$

$$y_{kv} \in \{0, 1\}, \quad \forall k \in K, v \in V^k, \quad (13)$$

$$z_v \geq 0, \quad \forall v \in V, \quad (14)$$

$$\phi_v \geq 0, \quad \forall v \in V, \quad (15)$$

$$b_v \geq 0, \quad \forall v \in V. \quad (16)$$

A função objetivo (1) minimiza o custo total, composto por frete por peso transportado, frete morto, e custos de múltiplas entregas. As restrições (2) garantem que cada item seja designado para um único veículo. As restrições (3) garantem que as capacidades dos veículos não sejam excedidas. As restrições (4) implicam que um veículo tem que visitar um cliente para entregar seus itens. As restrições (5) garantem que entregas incompatíveis não sejam atribuídas ao mesmo veículo. As restrições (6) determinam que um cliente só pode ser visitado por um veículo se este for utilizado. As restrições (7) definem o número de entregas adicionais na solução. As restrições (8) indicam que os itens atribuídos a um veículo mais o frete morto garantem a carga mínima sempre que este veículo for utilizado. As restrições (9) limitam a quantidade de frete morto. As restrições (10) estabelecem a tarifa de frete morto forçando ϕ_v a ser maior ou igual ao maior custo de entrega considerando os clientes visitados por aquele veículo (i.e., aqueles para os quais $y_{kv} = 1$). As restrições (11)-(13) asseguram a integralidade, enquanto as restrições (14)-(16) garantem os requisitos de não negatividade das variáveis.

5. Experimentos computacionais preliminares

Os experimentos foram executados em uma máquina rodando Ubuntu x86-64 GNU/Linux, com processador Intel Core i7-10700 Octa-Core 2.90 Ghz e 16Gb de RAM. A formulação foi implementada em Julia utilizando o pacote JuMP e resolvida com Gurobi 10.0.1. Foi estabelecido um tempo limite de 3600 segundos (uma hora) e foram adotadas as configurações padrão do Gurobi, com exceção do paralelismo que foi desativado.

5.1. Instâncias

Geramos um conjunto de instâncias sintéticas com $|I| \in \{25, 50, 75, 100, 150, 200\}$. Foram geradas três distribuições diferentes para as localizações dos clientes, todas dentro de uma grade de 100x100, com o remetente k_0 localizado no ponto (0, 0). A Figura 1 ilustra cada uma das distribuições. Na primeira distribuição, chamada **randômica** (rand) e exemplificada na Figura 1(a), as posições dos clientes são definidas aleatoriamente. Na segunda distribuição, denotada por **clusterizada uniforme** (unif) e exemplificada na Figura 1(b), os clientes foram agrupados em *clusters* de tamanhos quase iguais. Na última distribuição, denotada por **clusterizada não uniforme** (n_unif) e exemplificada na Figura 1(c), os clientes foram agrupados em *clusters* de diferentes tamanhos. A quantidade de *clusters* assumiu valores $C \in \{3, 5, 10\}$. Os demais parâmetros configuráveis considerados foram: $w_i \in [2, 15]$, $t_i \in \{1, 2\}$, $\tau \in \{1, 2\}$, $C_v \in \{25, 30, 35, 40, 45, 50\}$, $r_v = 50$, $maxd_v \in \{10, 20\}$, $C_v^{\min} = C_v \times f$, com $f \in \{0.85, 0.9, 0.95\}$, $p_v \in \{1.1, 1.21\}$ e $|I_k| \in [1, 5]$ o conjunto de itens do cliente k .

Os itens foram gerados com pesos aleatórios, e para cada um deles foi determinado um cliente aleatório junto com sua respectiva posição. Por fim, os veículos foram gerados utilizando uma heurística gulosa. Especificamente, cada um dos itens é analisado para inserção em algum veículo. Se houver veículo disponível para sua inserção, ele é colocado naquele de menor capacidade

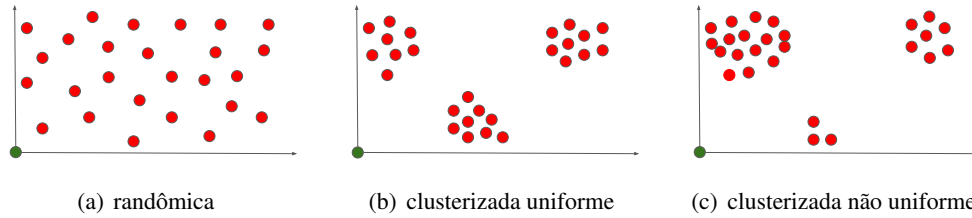


Figura 1: Exemplos de posicionamentos para 25 clientes (vermelho) e o remetente (verde) considerando as três distribuições.

disponível. Caso contrário, um novo veículo é criado selecionando aleatoriamente uma capacidade dentre as disponíveis e uma transportadora que possa atender o respectivo cliente. Foram consideradas cinco possibilidades de transportadoras, que se diferenciam pelos quadrantes da grade que podem atender e custos associados. No total, foram geradas 54 instâncias. As instâncias são identificadas por $D_|I|_C_ID$, em que D indica o tipo de distribuição, $|I|$ a quantidade de itens, C a quantidade de *clusters*, e ID um identificador da instância. A exceção é para as instâncias randômicas, que não apresentam um valor correspondente a C por não possuírem *clusters*.

5.2. Resultados

As Tabelas 1-3 resumem os resultados obtidos para cada tipo de instância gerada. A primeira coluna identifica a instância, a segunda coluna apresenta o valor da melhor solução encontrada, a terceira (*lb*) o limite inferior ao fim da execução de cada instância. A quarta coluna indica o tempo de execução em segundos. A quinta coluna, apresenta o *gap* de otimalidade em porcentagem, calculado como $100 \times \frac{|melhor-lb|}{melhor}$. A última coluna (*carga*(%)) mostra a média do percentual da capacidade utilizada dos veículos selecionados.

Tabela 1: Resultados da formulação para as instâncias randômicas

instância	melhor	lb	tempo (s)	gap (%)	carga (%)
rand_25_A	12917.8	12917.8	0.1	0.0	67.4
rand_25_B	6812.2	6812.2	1.9	0.0	76.6
rand_25_C	6670.9	6670.9	0.0	0.0	62.8
rand_50_A	16287.4	16287.4	146.5	0.0	65.6
rand_50_B	14380.9	14380.9	134.7	0.0	74.5
rand_50_C	19789.9	19789.9	1837.4	0.0	70.4
rand_75_A	22901.2	22163.5	3600.0	3.2	78.9
rand_75_B	21069.8	18513.7	3600.0	12.1	87.6
rand_75_C	28300.6	27451.2	3600.0	2.9	71.7
rand_100_A	28043.5	23690.4	3600.0	15.5	90.4
rand_100_B	35235.0	33047.8	3600.0	6.2	89.9
rand_100_C	32235.8	30301.9	3600.0	5.9	73.9
rand_150_A	127928.4	41365.6	3600.0	16.1	84.4
rand_150_B	42658.7	36912.4	3600.0	13.5	91.3
rand_150_C	27023.5	19908.1	3600.0	26.3	86.0
rand_200_A	47334.7	43770.5	3600.0	7.5	94.6
rand_200_B	56415.3	48984.0	3600.0	13.2	91.4
rand_200_C	54963.3	48199.3	3600.0	12.3	85.1
Média	33387.1	26175.9	2517.8	7.4	80.1

De maneira geral, os resultados mostram que as instâncias impõem um grande desafio para a formulação conseguir provar a otimalidade. O ótimo foi provado para 13 instâncias, sendo 9 com 25 itens e 4 com 50 itens. A distribuição randômica implica um espaçamento maior entre os

Tabela 2: Resultados da formulação para as instâncias clusterizadas uniformes

instância	melhor	lb	tempo (s)	gap (%)	carga (%)
unif_25_3_A	11018.2	11018.2	0.5	0.0	76.6
unif_25_5_A	3870.7	3870.7	0.1	0.0	57.3
unif_25_10_A	8096.3	8096.3	0.1	0.0	75.1
unif_50_3_A	16741.1	15436.8	3600.0	7.7	92.0
unif_50_5_A	19293.4	16383.4	3600.0	15.0	87.4
unif_50_10_A	12713.2	12713.2	176.9	0.0	83.6
unif_75_3_A	25178.0	24539.7	3600.0	2.5	96.8
unif_75_5_A	25251.1	20756.7	3600.0	17.7	87.9
unif_75_10_A	12214.4	11636.4	3600.0	4.7	87.5
unif_100_3_A	41973.6	39972.8	3600.0	4.7	94.6
unif_100_5_A	29771.4	27722.7	3600.0	6.8	96.7
unif_100_10_A	28669.8	24533.3	3600.0	14.4	92.1
unif_150_3_A	58469.5	56722.3	3600.0	2.9	97.8
unif_150_5_A	41805.6	32443.0	3600.0	22.4	93.9
unif_150_10_A	55398.8	49721.1	3600.0	10.2	92.4
unif_200_3_A	71389.5	67491.5	3600.0	5.4	94.6
unif_200_5_A	51595.1	46983.5	3600.0	8.9	95.5
unif_200_10_A	72600.2	68986.5	3600.0	4.9	94.6
Média	32558.3	29946.0	2809.8	7.1	88.6

Tabela 3: Resultados da formulação para as instâncias clusterizadas não uniformes

instância	melhor	lb	tempo (s)	gap (%)	carga (%)
n_unif_25_3_A	8320.6	8320.6	0.2	0.0	83.2
n_unif_25_5_A	7877.8	7877.8	5.3	0.0	69.6
n_unif_25_10_A	6090.0	6090.0	0.0	0.0	74.1
n_unif_50_3_A	24178.9	24061.7	3600.0	0.4	92.9
n_unif_50_5_A	22187.3	21481.2	3600.0	3.1	85.4
n_unif_50_10_A	21446.5	21265.7	3600.0	0.8	78.9
n_unif_75_3_A	21412.0	20750.7	3600.0	3.0	96.7
n_unif_75_5_A	10315.0	10083.1	3600.0	2.2	88.2
n_unif_75_10_A	13078.9	12971.0	3600.0	0.8	83.8
n_unif_100_3_A	22031.5	21159.7	3600.0	3.9	97.2
n_unif_100_5_A	35760.6	33542.6	3600.0	6.2	94.2
n_unif_100_10_A	38669.2	37409.1	3600.0	3.2	86.8
n_unif_150_3_A	67301.1	65435.1	3600.0	2.7	94.8
n_unif_150_5_A	67589.5	65774.2	3600.0	2.6	94.3
n_unif_150_10_A	41074.2	38228.1	3600.0	6.9	91.1
n_unif_200_3_A	65847.3	61633.9	3600.0	6.4	96.5
n_unif_200_5_A	78652.8	73803.8	3600.0	6.1	94.7
n_unif_200_10_A	79886.9	74739.8	3600.0	6.4	93.1
Média	35095.5	33590.4	3000.3	3.0	88.6

clientes. A Tabela 1 indica que, conseqüentemente, é necessário gerar mais veículos do que para as outras distribuições, pois a distância impede em muitos casos a reentrega, diminuindo o número de restrições da formulação. Entretanto, isso leva a um aproveitamento menos vantajoso da capacidade dos veículos, ficando abaixo de 75% nas instâncias randômicas com até 50 itens.

A Tabela 2 mostra uma tendência em que quanto menor o número de centros, menor o *gap* final para as instâncias clusterizadas uniformes. Vale ressaltar que não existe correspondência entre as posições dos *clusters* nas instâncias clusterizadas uniformes e não uniformes. Por isso, não se pode fazer uma comparação direta quanto aos custos entre os dois tipos de instâncias que tenham a mesma quantidade de itens e de centros. Conseqüentemente, isso faz com que o aumento do número

de centros não seja um fator determinante forte o suficiente para ser considerado isoladamente. Vale ressaltar que quanto menor o valor da relação *clientes/centros* mais as instâncias clusterizadas uniformes e não uniformes se aproximam da distribuição randômica. Por exemplo, a instância *unif_50_10_A* tem o valor da relação *clientes/centros* de 1.4 e tem o *gap* mais baixo para esse grupo uniforme com 50 itens. Claro que esse não é o único fator determinante para a dificuldade de uma instância.

A Tabela 3 mostra que as instâncias clusterizadas não uniformes tiveram o menor *gap* médio e muito por conta do que já foi citado anteriormente quanto a um comportamento que em alguns casos aproxima muitos *clusters* de uma distribuição randômica e poucos que permitem uma quantidade maior de reentrega. Ainda assim, é possível ter um bom aproveitamento da capacidade dos veículos.

Comparando as três tabelas, pode-se notar que as instâncias randômicas parecem ser mais fáceis quando o número de itens é pequeno. Além disso, neste mesmo cenário, as instâncias clusterizadas uniformes e não uniformes tendem a se assemelhar a uma instância randômica, principalmente com o aumento do número de centros. Em contrapartida, existe muito mais capacidade não aproveitada nas randômicas, enquanto o aproveitamento é bem melhor nas que contém estruturas de *clusters*, pois a concentração de clientes permite mais reentregas.

Contudo, para a distribuição randômica, o aumento do número de itens e do número de clientes aumenta as chances da distância entre dois clientes permitir reentregas e ter um melhor aproveitamento da capacidade dos veículos utilizados. Esse caso fica claro com as instâncias *rand_150_C* (Figura 2(a)) e *rand_200_B* (Figura 2(b)). Essas duas instâncias possuem, respectivamente, 3 e 8 clientes a mais que veículos. Por outro lado, nas de 50 itens a quantidade de veículos é maior ou igual à quantidade de clientes. Outro fato a ser notado é que existe um aumento do aproveitamento dos veículos utilizados à medida em que o número de itens e clientes aumenta.

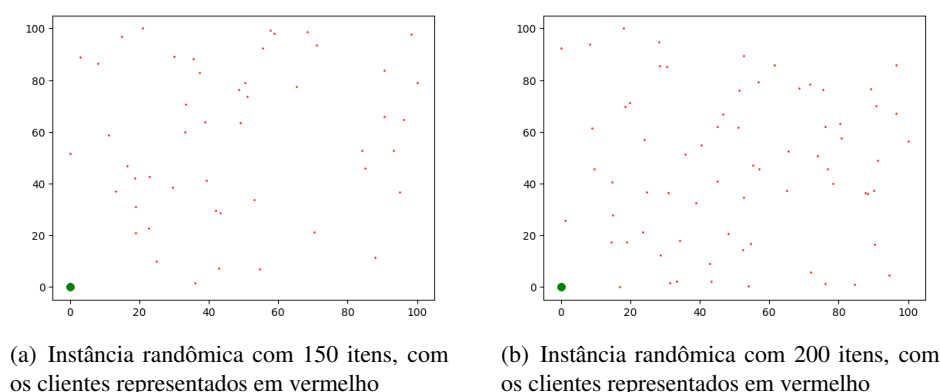
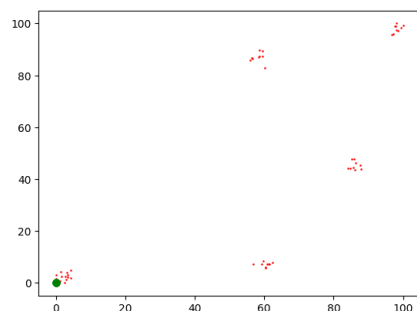
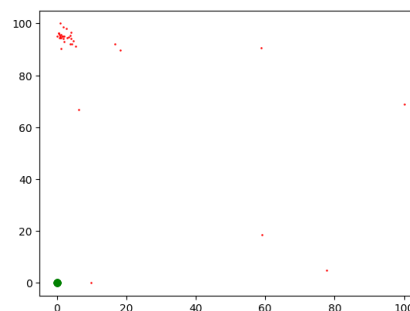


Figura 2: Exemplos de instâncias randômicas

A Figura 3(a) representa a instância *unif_150_5_A*, que ilustra como outros fatores influenciam. A existência de uma maior concentração de clientes aumenta as possibilidades de reentrega, mas também existe um *cluster* perto da origem cujo custo de entrega é mais baixo o que aumenta as possibilidades da utilização de veículos para atender um único cliente. Ainda assim, o aproveitamento da capacidade dos veículos está acima de 93%. Essa análise fica um pouco mais complicada para as clusterizadas não uniformes, pois uma instância muito desbalanceada pode apresentar um comportamento similar ao de um único grande *cluster*. A Figura 3(b) mostra a instância *n_unif_100_10_A*, na qual existe um grande *cluster* e os demais terminam ficando muito isolados como em uma instância randômica. Consequentemente, em 8 desses 10 *clusters* não existe



(a) Instância clusterizada uniforme com 5 centros e 150 itens, com os clientes representados em vermelho



(b) Instância clusterizada não uniforme com 10 centros e 100 itens, com os clientes representados em vermelho

Figura 3: Exemplos de instâncias clusterizadas uniformes e não uniformes

a possibilidade de reentrega e em um deles apenas uma possibilidade, o que simplifica o modelo final gerado. Então o desbalanceamento é um fator importante, mas não o único determinante para facilitar ou dificultar a formulação encontrar a solução ótima.

6. Considerações finais

Neste artigo, propusemos e formalizamos um problema de consolidação de cargas com frota heterogênea terceirizada e custos relacionados a frete morto e múltiplas reentregas. Demonstramos que mesmo uma versão restrita desse problema é NP-difícil. Como método de solução, apresentamos uma formulação de programação inteira. Além disso, desenvolvemos um gerador de instâncias parametrizável que permite criar casos de testes com diferentes tipos de distribuições de localização de clientes.

Os experimentos computacionais mostraram que a formulação encontra a solução ótima para todas as instâncias com até 25 itens e quatro com 50 itens. Outro resultado identificado é que a formulação tem mais facilidade em resolver instâncias que contenham *clusters* desde que esses estejam bem definidos, que seria acima de 100 itens, utilizando melhor as capacidades dos veículos, apesar de não ser o único fator determinante. A partir dessa quantidade de itens, o *gap* médio é de 12.9%, 8.9% e 4.9% respectivamente para as randômicas, uniformes e não uniformes. Também foi possível notar que mesmo nas uniformes é necessário haver uma razão mais alta entre *clientes/centros* para que as estruturas de *clusters* sejam bem caracterizadas e não se aproximem de uma distribuição randômica. Por fim, foi visto que a distribuição randômica leva a um pior aproveitamento da capacidade dos veículos em comparação à uniforme e à não uniforme.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001. Este trabalho foi parcialmente apoiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), processos 314662/2020-0 e 314718/2023-0. Este trabalho foi parcialmente apoiado pelo projeto FAPESB INCITE PIE0002/2022.

Referências

Christiansen, M., Fagerholt, K., e Ronen, D. (2004). Ship routing and scheduling: Status and perspectives. *Transportation Science*, 38(1):1–18.

- Ekici, A. (2022). Variable-sized bin packing problem with conflicts and item fragmentation. *Computers & Industrial Engineering*, 163:107844.
- Gendreau, M., Laporte, G., e Semet, F. (2004). Heuristics and lower bounds for the bin packing problem with conflicts. *Computers & Operations Research*, 31(3):347–358.
- Hajiaghaei-Keshteli, M., J Afshari, A., e Nasiri, E. (2017). Addressing the freight consolidation and containerization problem by recent and hybridized meta-heuristic algorithms. *International Journal of Engineering*, 30(3):403–410.
- Hanbazazah, A. S., Abril, L., Erkoc, M., e Shaikh, N. (2019). Freight consolidation with divisible shipments, delivery time windows, and piecewise transportation costs. *European Journal of Operational Research*, 276(1):187–201.
- IBGE, A. N. (2014). IBGE mapeia a infraestrutura dos transportes no Brasil. <https://agenciadenoticias.ibge.gov.br/agencia-sala-de-imprensa/2013-agencia-de-noticias/releases/14707-asi-ibge-mapeia-a-infraestrutura-dos-transportes-no-brasil>. Acessado em: 06/05/2024.
- Jansen, K. (1999). An approximation scheme for bin packing with conflicts. *Journal of Combinatorial Optimization*, 3(4):363–377.
- Melo, R. A. e Ribeiro, C. C. (2015). Improved solutions for the freight consolidation and containerization problem using aggregation and symmetry breaking. *Computers & Industrial Engineering*, 85:402–413.
- Nataraj, S., Ferone, D., Quintero-Araujo, C., Juan, A., e Festa, P. (2019). Consolidation centers in city logistics: A cooperative approach based on the location routing problem. *International Journal of Industrial Engineering Computations*, 10(3):393–404.
- Qin, H., Zhang, Z., Qi, Z., e Lim, A. (2014). The freight consolidation and containerization problem. *European Journal of Operational Research*, 234(1):37–48.
- Sadykov, R. e Vanderbeck, F. (2013). Bin packing with conflicts: a generic branch-and-price algorithm. *INFORMS Journal on Computing*, 25(2):244–255.
- Tiwari, S., Wee, H. M., Zhou, Y., e Tjoeng, L. (2021). Freight consolidation and containerization strategy under business as usual scenario & carbon tax regulation. *Journal of Cleaner Production*, 279:123270.
- Vieira, T. e Munari, P. (2021). A MIP model for freight consolidation in road transportation considering outsourced fleet. In *Optimization and Data Science: Trends and Applications*, p. 99–109. Springer.