LI Simpósio Brasileiro de Pesquisa Operacional
Limeira–SP, 02 a 06 de setembro de 2019

Um algoritmo híbrido para o problema de roteamento de estoques com produto perecível

Aldair Alvarez
Departamento de Engenharia de Produção, Universidade Federal de São Carlos
aldair@dep.ufscar.br

Jean-François Cordeau, Raf Jans
Departamento de Logística e Gestão de Operações, HEC Montréal
jean-francois.cordeau@hec.ca, raf.jans@hec.ca

Pedro Munari, Reinaldo Morabito
Departamento de Engenharia de Produção, Universidade Federal de São Carlos
munari@dep.ufscar.br, morabito@ufscar.br

RESUMO

Neste trabalho, aborda-se uma variante do problema de roteamento de estoques em que um único fornecedor é responsável pela entrega de um produto perecível a um conjunto de clientes, ao longo de um horizonte de planejamento multiperíodo. Assume-se que o produto tenha uma vida útil pré-determinada, durante a qual ele é utilizável, devendo ser descartado imediatamente após esse período. Para resolver este problema, é proposto um algoritmo híbrido baseado na combinação de uma metaheurística de busca local iterada com uma formulação de programação linear. A efetividade do método é testada usando instâncias da literatura, assim como novas instâncias do problema propostas neste trabalho.

PALAVRAS CHAVE. Roteamento de estoques. Perecibilidade. Método híbrido.

L&T-Logística e Transportes; MH-Metaheurísticas

ABSTRACT

In this paper we address a variant of the inventory routing problem in which a single supplier is responsible for delivering a single perishable product to a set of customers, over a multiperiod planning horizon. It is assumed that the product has a fixed shelf-life during which it is usable and after which it should be discarded. To solve this problem, we propose a hybrid algorithm based on the combination of an iterated local search metaheuristic with a linear programming formulation. The effectiveness of the method is tested using instances of the literature as well as new problem instances proposed in this work.

KEYWORDS. Inventory routing. Perishability. Hybrid method.

L&T-Logistics and transportation; MH-Metaheuristics

https://proceedings.science/p/106697?lang=pt-br

1. Introdução

Nas últimas décadas, tem-se notado um crescimento considerável de pesquisas a respeito dos ganhos que se obtém com a integração de múltiplas atividades ao longo da cadeia de suprimentos. Como resultado, sabe-se atualmente que essa integração pode levar a vantagens significativas em termos econômicos e de desempenho. Esse é o caso do problema de roteamento de estoques (PRE), o qual integra a gestão de estoques com o roteamento de veículos, muito comum em abordagens em que o estoque do cliente é gerenciado pelo fornecedor. Por outro lado, tais integrações podem resultar em problemas complexos e difíceis de serem resolvidos na prática.

No contexto do PRE, é de particular relevância considerar a perecibilidade dos produtos, dado que em muitas indústrias tem-se que as matérias primas, bem como produtos intermediários e finais, podem ser perecíveis. Além disso, a perecibilidade pode aparecer mais de uma vez ao longo da cadeia de suprimentos. Portanto, o gerenciamento da perecibilidade se torna um assunto relevante na cadeia de suprimentos, particularmente nas atividades de controle de estoques.

A perecibilidade no contexto dos PREs foi estudada pela primeira vez por Federgruen et al. [1986], que abordaram um problema de gestão de estoques e distribuição para um produto que devia ser descartado caso não fosse usado durante uma vida útil pré-determinada. Os autores estudaram diferentes padrões e políticas para a parte de distribuição do problema. O objetivo consistiu em minimizar a soma dos custos de transporte, de falta do produto e de descarte. Desde então, diversas pesquisas vêm abordando aspectos relacionados à perecibilidade dos produtos no contexto do PRE [Hemmelmayr et al., 2009; Le et al., 2013; Coelho e Laporte, 2014; Mirzaei e Seifi, 2015; Diabat et al., 2016; Azadeh et al., 2017; Crama et al., 2018; Rohmer et al., 2019].

Neste trabalho, aborda-se o PRE para um único produto perecível (PREP), conforme proposto por Coelho e Laporte [2014]. Nesse problema, a perecibilidade é modelada pela definição de uma vida útil dentro da qual o produto é usável e após a qual o produto deve ser descartado. Além disso, são definidos custos de estocagem e receitas de venda dependentes da idade do produto. Para resolver este problema, é proposto um algoritmo híbrido baseado na combinação de uma metaheurística de busca local iterada (BLI) com uma formulação de programação linear (PL). A efetividade do método é testada usando instâncias da literatura, assim como novas instâncias para o problema.

As demais seções deste trabalho estão organizadas da seguinte maneira. Na Seção 2, descreve-se formalmente o PREP e apresenta-se uma formulação matemática para o problema, juntamente com um método *branch-and-cut* para resolvê-la. Depois, na Seção 3, é apresentado o método híbrido desenvolvido para resolver o problema. Finalmente, nas Seções 4 e 5, são apresentados os experimentos computacionais e as principais conclusões desta pesquisa.

2. Descrição do problema e métodos de solução

No PREP [Coelho e Laporte, 2014], um fornecedor é responsável pela entrega de um único produto perecível a um conjunto de clientes ao longo de um horizonte de planejamento finito. Assume-se que o produto tenha uma vida útil fixa, durante a qual é utilizável, devendo ser imediatamente descartado após esse período.

O problema pode ser definido em um grafo completo não direcionado $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, onde $\mathcal{N} = \{0, 1, \ldots, N\}$ é o conjunto de vértices e $\mathcal{E} = \{(i, j) \colon i, j \in \mathcal{N}, i < j\}$ é o conjunto de arestas. O vértice 0 representa o depósito do fornecedor, o qual possui uma frota homogênea de K veículos de capacidade Q, sendo representada pelo conjunto $\mathcal{K} = \{1, \ldots, K\}$. Os demais vértices do conjunto $\mathcal{N}$, denotados por $\mathcal{C} = \{1, \ldots, N\}$, representam os clientes. O horizonte de tempo é denotado pelo conjunto $\mathcal{T} = \{1, \ldots, T\}$ de períodos de tempo.

O produto perecível se torna inutilizável S períodos de tempo após ser disponibilizado no depósito do fornecedor, sendo que a idade do produto aumenta em uma unidade a cada período de tempo. Assim, a idade do produto pertence a um conjunto $\mathcal{S} = \{0, 1, \ldots, S\}$ e é usada para definir seu valor, de acordo com a receita de vendas u_{is} especificada para cada unidade do produto com idade $s \in \mathcal{S}$ (usada para satisfazer a demanda) pelo cliente $i \in \mathcal{C}$. Um custo de viagem c_{ij} é associado a cada aresta $(i, j) \in \mathcal{E}$ e um custo de estocagem dependente da idade h_{is} é cobrado tanto no fornecedor 0 como nos clientes $i \in \mathcal{C}$ por cada unidade estocada do produto de idade $s \in \mathcal{S}$ no final de cada período de tempo. Como em Coelho e Laporte [2014], considera-se que o produto que atingiu sua idade máxima ($s = S$) no final de um período de tempo, não irá para a área de armazenamento regular, mas será mantido separadamente no estoque para ser descartado no próximo período. Assim, essas quantidades incorrem nos custos de estocagem, mas não limitam a quantidade que o cliente pode receber no próximo período de tempo.

Cada cliente $i \in \mathcal{C}$ tem uma capacidade de estocagem limitada C_i e para cada nó (fornecedor e os clientes) $i \in \mathcal{N}$ tem-se um estoque inicial I_{i0}^0 de produto fresco (idade 0) disponível no início do horizonte de planejamento ($t = 0$). Assim, o estoque inicial terá idade 1 no primeiro período de tempo do horizonte de planejamento ($t = 1$). Cada cliente $i \in \mathcal{C}$ tem uma demanda conhecida d_i^t em cada período de tempo $t \in \mathcal{T}$. Adicionalmente, o fornecedor recebe uma quantidade r^t de produto fresco (idade 0) em cada período de tempo $t \in \mathcal{T}$. No entanto, esta quantidade está disponível para entrega apenas um período de tempo depois de ficar disponível no depósito do fornecedor.

O PREP consiste em determinar os períodos de tempo em que os clientes serão visitados; a quantidade de produto de cada idade disponível que será entregue em cada visita aos clientes; a quantidade de produto de cada idade disponível que será usada para satisfazer a demanda de cada cliente; e as rotas de entrega que serão realizadas para fazer as entregas. O objetivo é maximizar o lucro total, dado pela receita das vendas menos a soma dos custos de estocagem e de transporte. Os custos de estocagem são cobrados nos níveis de estoque no final de cada período de tempo, tanto no fornecedor quanto nos clientes. Considera-se que os produtos de diferentes idades compartilham o mesmo espaço de armazenamento em todos os nós. Também, assume-se que a capacidade de armazenamento do fornecedor é ilimitada. Além disso, de acordo com a prática da literatura, admite-se que os clientes que recebem uma entrega em um determinado período de tempo podem usá-la para atender a demanda no mesmo período de tempo.

2.1. Modelagem matemática

Coelho e Laporte [2014] propõem modelar o PREP matematicamente por meio de uma formulação baseada no fluxo nas arestas de um grafo. Para isso, a seguinte notação é introduzida. Seja $\mathcal{S}^t = \{s \in \mathcal{S} \colon 1 \leq s \leq t\}$ o subconjunto de idades do produto que estão disponíveis para entrega em cada período de tempo t. Este conjunto também especifica as idades que podem ser usadas para satisfazer a demanda do cliente no período de tempo determinado. Note que este conjunto não contém a idade 0, a qual faz parte do conjunto de idades $\mathcal{S}$ e que está disponível no fornecedor em cada período de tempo, pois o produto de idade 0 disponibilizado no fornecedor em um determinado período, só pode ser usado para entregas no período seguinte. Finalmente, considere as seguintes variáveis de decisão:

$x_{ij}^{kt} \in \{0, 1, 2\}$: número de vezes que o veículo k percorre a aresta (i, j) no período de tempo t,
$y_i^{kt} \in \{0, 1\}$: 1 se o nó i é visitado pelo veículo k no período de tempo t, 0 caso contrário;
$I_{is}^t \geq 0$: nível de estoque do produto de idade s no nó i no final do período de tempo t;
$q_{is}^{kt} \geq 0$: quantidade do produto de idade s entregue ao cliente i pelo veículo k

no período de tempo t;

$w_{is}^t \geq 0$: quantidade do produto de idade s usada para satisfazer a demanda do cliente i no período de tempo t.

Usando essas variáveis, uma formulação matemática do problema pode ser definida como:

$$\max \quad \sum_{i\in\mathcal{C}}\sum_{t\in\mathcal{T}}\sum_{s\in\mathcal{S}^t} u_{is}w_{is}^t - \sum_{(i,j)\in\mathcal{E}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} c_{ij}x_{ij}^{kt} - \sum_{i\in\mathcal{N}}\sum_{t\in\mathcal{T}}\sum_{s\in\mathcal{S}^t} h_{is}I_{is}^t - \sum_{t\in\mathcal{T}} h_{00}r^t \tag{1}$$

$$\text{s.t.} \quad I_{0s}^t = r^t \qquad t\in\mathcal{T}, s=0, \tag{2}$$

$$I_{0s}^t = I_{0,s-1}^{t-1} - \sum_{i\in\mathcal{C}}\sum_{k\in\mathcal{K}} q_{is}^{kt} \qquad t\in\mathcal{T}, s\in\mathcal{S}^t, \tag{3}$$

$$I_{is}^t = I_{i,s-1}^{t-1} + \sum_{k\in\mathcal{K}} q_{is}^{kt} - w_{is}^t \qquad i\in\mathcal{C}, t\in\mathcal{T}, s\in\mathcal{S}^t, \tag{4}$$

$$d_i^t = \sum_{s\in\mathcal{S}^t} w_{is}^t \qquad i\in\mathcal{C}, t\in\mathcal{T}, \tag{5}$$

$$\sum_{s\in\mathcal{S}^{t-1}\setminus\{S\}} I_{is}^{t-1} + \sum_{k\in\mathcal{K}}\sum_{s\in\mathcal{S}^t} q_{is}^{kt} \leq C_i \qquad i\in\mathcal{C}, t\in\mathcal{T}, \tag{6}$$

$$\sum_{s\in\mathcal{S}^t} q_{is}^{kt} \leq \min\{Q, C_i\} y_i^{kt} \qquad i\in\mathcal{C}, k\in\mathcal{K}, t\in\mathcal{T}, \tag{7}$$

$$\sum_{i\in\mathcal{C}}\sum_{s\in\mathcal{S}^t} q_{is}^{kt} \leq Q y_0^{kt} \qquad k\in\mathcal{K}, t\in\mathcal{T}, \tag{8}$$

$$\sum_{j\in\mathcal{N}:j<i} x_{ji}^{kt} + \sum_{j\in\mathcal{N}:j>i} x_{ij}^{kt} = 2y_i^{kt} \qquad i\in\mathcal{N}, k\in\mathcal{K}, t\in\mathcal{T}, \tag{9}$$

$$\sum_{i\in\mathcal{B}}\sum_{j\in\mathcal{B}:j>i} x_{ij}^{kt} \leq \sum_{i\in\mathcal{B}\setminus\{l\}} y_i^{kt} \qquad \forall\mathcal{B}\subseteq\mathcal{C}, |\mathcal{B}|\geq 2, k\in\mathcal{K}, t\in\mathcal{T}, l\in\mathcal{B}, \tag{10}$$

$$\sum_{k\in\mathcal{K}} y_i^{kt} \leq 1 \qquad i\in\mathcal{C}, t\in\mathcal{T}, \tag{11}$$

$$I_{is}^t \geq 0 \qquad i\in\mathcal{N}, t\in\mathcal{T}, s\in\mathcal{S}^t, \tag{12}$$

$$q_{is}^{kt} \geq 0 \qquad i\in\mathcal{C}, k\in\mathcal{K}, t\in\mathcal{T}, s\in\mathcal{S}^t, \tag{13}$$

$$w_{is}^t \geq 0 \qquad i\in\mathcal{C}, t\in\mathcal{T}, s\in\mathcal{S}^t, \tag{14}$$

$$y_i^{kt} \in \{0,1\} \qquad i\in\mathcal{N}, k\in\mathcal{K}, t\in\mathcal{T}, \tag{15}$$

$$x_{ij}^{kt} \in \{0,1\} \qquad (i,j)\in\mathcal{E}: i\neq 0, k\in\mathcal{K}, t\in\mathcal{T}, \tag{16}$$

$$x_{ij}^{kt} \in \{0,1,2\} \qquad (i,j)\in\mathcal{E}: i=0, k\in\mathcal{K}, t\in\mathcal{T}. \tag{17}$$

A função objetivo (1) consiste em maximizar a lucro total, dado pela receita de vendas menos a soma dos custos de transporte e estocagem. O último termo da função objetivo corresponde ao custo de estocagem incorrido pela quantidade de produto fresco disponibilizado no fornecedor em cada período de tempo. As restrições (2) e (3) definem o balanço do estoque no fornecedor para as idades 0 e no conjunto $\mathcal{S}^t$, respectivamente. O conjunto de restrições (4) define o balanço do estoque para os clientes. As restrições (5) garantem o atendimento da demanda, o que pode ser feito usando produtos de diferentes idades. Nas restrições (6) impõe-se o nível de estoque

máximo após as entregas em cada cliente. Observe que produtos com idade S ainda disponíveis no final do período $t-1$ não entram no espaço de armazenamento regular e, portanto, não limitam a quantidade que pode ser entregue no período t. As restrições (7) permitem a entrega a um cliente por um veículo só se ele for visitado pelo respectivo veículo. As restrições (8) garantem que a capacidade de cada veículo é respeitada. As restrições (9) e (10) impõem a conservação do fluxo dos veículos e eliminam subciclos, respetivamente. O conjunto de restrições (11) limita o número de visitas que cada cliente pode receber em cada período de tempo. Finalmente, o domínio das variáveis é definido nas restrições (12)–(17).

2.2. Método *branch-and-cut*

A formulação (1)–(17) contém um número exponencial de restrições de eliminação de subciclo (10), o que requer o uso de um algoritmo do tipo *branch-and-cut* para resolvê-la. Assim, essas restrições são adicionadas de forma iterativa na árvore de busca. Para identificar as restrições violadas, utiliza-se um algoritmo de separação exato, que resolve uma série de problemas de corte mínimo. Esse algoritmo é aplicado para cada veículo e cada período de tempo. Em cada nó da árvore de busca, seja $\bar{y}_i^{kt}$ e $\bar{x}_{ij}^{kt}$ o valor das variáveis de visita (y) e fluxo (x) na solução, respectivamente. Um grafo é construído para cada veículo k e período t para os nós com $\bar{y}_i^{kt} > 0$, definindo os pesos das arestas do novo grafo como $\bar{x}_{ij}^{kt}$, $\forall (i,j) \in \mathcal{E}$. Depois, para cada nó correspondente a clientes, resolve-se um problema de corte mínimo, definindo a origem do grafo como o depósito (0) e o cliente em questão como o destino. Uma restrição violada é identificada se a capacidade do corte mínimo for menor que $2\bar{y}_i^{kt}$. Se um subciclo é encontrado em um conjunto de clientes $\mathcal{B} \subseteq \mathcal{C}$ com o veículo k no período t, as restrições (10) são adicionadas para o nó $l = \arg\max_{i \in \mathcal{B}}\{\bar{y}_i^{kt}\}$, para todos os veículos e períodos de tempo do horizonte de planejamento. Para resolver os problemas de corte mínimo, pode-se recorrer ao solver Concorde, conforme descrito em Adulyasak et al. [2014].

3. Método híbrido baseado em busca local iterada

Nessa seção, é proposto um algoritmo híbrido para o PREP, baseando-se na metaheurística BLI originalmente proposta por Alvarez et al. [2018] para a variante básica do PRE. Além disso, o algoritmo inclui um componente de programação matemática em sua estrutura. A ideia básica da BLI [Lourenço et al., 2010] é aplicar iterativamente um algoritmo de busca local a soluções resultantes da perturbação dos ótimos locais visitados anteriormente, o que leva a uma busca aleatória no espaço dos ótimos locais. O pseudocódigo do método proposto é mostrado no Algoritmo 1.

Algoritmo 1: Método híbrido baseado em busca local iterada

```
1  início
2      O^0 ← heurística_construtiva();
3      se O^0 ≠ ∅ então
4          O* ← busca_local (O^0);
5          enquanto stop criterion is not met faça
6              O' ← perturbação(O*);
7              O' ← busca_local(O');
8              O' ← otimizar_PL(O');
9              se f(O') > f(O*) então O* ← O' ;
10         fim
11     fim
12 fim
```

O algoritmo começa com uma solução inicial factível (linha 2), a qual é gerada usando a heurística de construção que será descrita na Seção 3.1. Se a heurística de construção não for

https://proceedings.science/p/106697?lang=pt-br

capaz de encontrar uma solução factível, o algoritmo é interrompido; caso contrário, o processo de busca continua. Uma heurística de busca em vizinhança variável com ordenação aleatória (RVND, do inglês *randomized variable neighborhood descent*) é usada como procedimento de busca local (linhas 4 e 7), e um algoritmo multi-operador é utilizado como um mecanismo de perturbação (linha 6). As variáveis contínuas q, w, e I (entregas, consumos e estoques) da solução são então otimizadas resolvendo uma formulação de fluxo *multicommodity* (FFM) (linha 8). O critério de aceitação admite a solução resultante somente se for melhor que a incumbente atual (linha 9). Todos os componentes do método híbrido são descritos em detalhes nas seguintes seções.

3.1. Heurística construtiva

Para obter uma solução factível para o problema, aplicou-se uma heurística de decomposição que separa iterativamente as decisões do problema em duas fases. Na primeira fase, a heurística define o tamanho das entregas potenciais aos clientes e atribui uma prioridade a cada um deles. Então, na segunda fase, rotas de entrega viáveis são construídas para tentar entregar as quantidades definidas na primeira fase.

A heurística começa utilizando o estoque inicial de cada cliente para atender a maior quantidade possível de demandas do próprio cliente. Depois, computa os níveis agregados de estoque (considerando os produtos de todas as idades). Esses níveis são atualizados no final de cada iteração de acordo com as entregas realizadas e os respectivos consumos. Além disso, o nível de estoque agregado é usado como guia para determinar o tamanho das entregas potenciais em cada iteração da heurística.

A heurística executa uma iteração para cada período de tempo $t \in \mathcal{T}$, iniciando em $t = 1$. Na primeira fase da iteração t, a heurística define uma quantidade de entrega potencial para cada cliente, calculando a diferença entre a capacidade e o nível de estoque agregado no final do período anterior, respeitando também a capacidade do veículo. Para simplificar a heurística, aplica-se uma abordagem gulosa na qual todas as entregas são do produto mais fresco possível, isto é, com idade $s = 1$. Portanto, a entrega potencial para cada cliente i é definida como $\min\{\texttt{ratio_demand} \times (C_i - I_i^{t-1}), Q\}$, onde $\texttt{ratio_demand} \in (0, 1]$ é um parâmetro que define a proporção da entrega máxima que de fato a heurística tentará entregar. A prioridade do cliente i é definida como o número vezes que a demanda do cliente ainda não foi satisfeita nos próximos `look_ahead` períodos (incluindo t). O valor de `look_ahead` determina quanto olhar para frente no horizonte de planejamento, tentando antecipar as próximas faltas do produto.

Depois de definir essas entregas e prioridades, a segunda fase da iteração t é iniciada. Essa fase consiste na determinação de uma ou mais rotas usando uma heurística de inserção do vizinho mais próximo que consiste em, num primeiro momento, incluir os clientes com maior prioridade desde que a inserção satisfaça a capacidade do veículo. No máximo, K rotas podem ser definidas nesta fase. Em seguida, considerando as entregas realmente realizadas, define-se os valores das variáveis de entrega e os valores das variáveis de consumo (usando a regra primeiro a entrar, primeiro a ser consumido). Depois os valores das variáveis de estoque são atualizados. Finalmente, uma nova iteração é iniciada para o próximo período ($t + 1$), até atingir o período de tempo T.

Dado que a heurística é executada em tempos computacionais relativamente curtos, ela foi definida dentro de dois ciclos externos, explorando diferentes valores para `ratio_demand` e `look_ahead`, com o objetivo de encontrar uma solução de melhor qualidade. Além disso, ao final da execução da heurística, as variáveis contínuas da melhor solução factível (se houver) são otimizadas usando a FFM da Seção 3.4.

https://proceedings.science/p/106697?lang=pt-br

3.2. Heurística de busca local

A busca local do método proposto neste trabalho é realizada por meio de uma heurística de busca de vizinhança variável [Mladenović e Hansen, 1997] com ordenação aleatória, sendo conhecida por RVND (do inglês *Randomized Variable Neighborhood Descent*). A heurística aplica repetidamente um conjunto de operadores de busca local na tentativa de melhorar progressivamente a solução incumbente. O funcionamento da heurística RVND é descrito a seguir. Dado um conjunto de operadores de busca local, a heurística escolhe aleatoriamente um deles para melhorar a solução atual. Se o operador não melhora a solução, ele é eliminado do conjunto de operadores e a busca continua com os outros operadores. Por outro lado, toda vez que um operador de busca local melhora a solução atual, o conjunto é restabelecido a sua forma inicial. O algoritmo termina quando o conjunto de operadores se torna vazio.

Neste trabalho, a fase de busca local é utilizada apenas para administrar as decisões relacionadas ao roteamento. Portanto, a heurística RVND usa os seguintes operadores clássicos do problema de roteamento de veículos: Or-opt-k, $k \in \{1, 2, 3\}$; Shift(k), $k \in \{1, 2, 3\}$; Swap(k_1, k_2), $k_1, k_2 \in \{1, 2\}, k_1 \geq k_2$; e k-opt, $k \in \{2, 3\}$. Apenas movimentos factíveis são admitidos e a estratégia de primeira melhoria é utilizada nos operadores de busca local.

3.3. Mecanismo de perturbação

Como no PREP há diferentes decisões que devem ser tomadas simultaneamente, foi desenvolvido um algoritmo de perturbação que pode alterar vários atributos de uma solução em uma única chamada. O algoritmo usa os seguintes operadores para modificar as decisões de visita e entrega de uma solução de entrada:

- *Inserir visitas*: uma rota e um cliente que não seja visitado pela rota são escolhidos aleatoriamente. O cliente é inserido na posição de inserção mais barata na rota. Então os valores das variáveis contínuas do problema são re-otimizados usando a FFM;

- *Remover visitas*: uma rota é escolhida aleatoriamente e, em seguida, um cliente aleatório da rota é removido. Depois disso, os valores das variáveis contínuas são re-otimizados usando a FFM;

- *Mover visita*: para uma rota aleatoriamente escolhida, seleciona-se um cliente da rota que não seja visitado em todos os períodos de tempo do horizonte de planejamento. Em seguida, a visita ao cliente é removida da rota e inserida na posição mais barata de uma rota de um período no qual o cliente não é visitado, ambos são escolhidos aleatoriamente (o novo período e a rota). Finalmente, os valores das variáveis contínuas são re-otimizados usando a FFM; e

- *Reduzir entregas*: escolha uma rota e uma entrega (de uma certa idade) de um cliente da rota tal que o total entregue não é completamente consumido pelo cliente. Isso pode acontecer, por exemplo, quando é lucrativo acumular estoque no cliente para economizar custos de estocagem no fornecedor. Em seguida, a entrega é reduzida pela quantidade não consumida pelo cliente. Tanto o cliente quanto a entrega a ser reduzida são escolhidos aleatoriamente.

Depois de aplicar cada operador, o valor da função objetivo da solução é recalculado. O objetivo desses operadores é ajudar a determinar os períodos nos quais cada cliente deve ser visitado e criar folgas nas rotas a fim de ajudar à heurística de busca local. Nos operadores *Remover visitas* e *Mover visitas*, as soluções infactíveis são rejeitadas. Nesse caso, o operador escolhe outro cliente da mesma rota. Se todos os clientes da rota escolhida forem explorados sem sucesso (resultando em soluções inviáveis), o operador escolhe outra rota e o processo é aplicado da mesma forma.

https://proceedings.science/p/106697?lang=pt-br

O parâmetro `max_perturb` define o número máximo de elementos de uma solução que podem ser alterados toda vez que o mecanismo de perturbação é chamado. Assim, semelhante à heurística RVND, nosso algoritmo de perturbação pode usar vários operadores em uma única chamada, aplicando um operador por vez (mudando no máximo um elemento da solução) até que o número de alterações executadas na solução atinja `max_perturb` ou nenhum dos operadores possa alterar a solução.

3.4. Formulação de fluxo *multicommodity* (FFM)

Dados os valores das variáveis de visita ($\bar{y}$) de uma solução, pode-se determinar os valores ótimos para as variáveis de entrega, consumo e estoque que maximizam o lucro total resolvendo a seguinte FFM:

$$\max \quad \sum_{i\in\mathcal{C}}\sum_{t\in\mathcal{T}}\sum_{s\in\mathcal{S}^t} u_{is}w_{is}^t - \sum_{i\in\mathcal{N}}\sum_{t\in\mathcal{T}}\sum_{s\in\mathcal{S}^t} h_{is}I_{is}^t \tag{18}$$

$$\text{s.t.} \quad \sum_{s\in\mathcal{S}^t} q_{is}^{kt} \leq \min\{Q, C_i\}\bar{y}_i^{kt} \qquad i\in\mathcal{C}, k\in\mathcal{K}, t\in\mathcal{T}, \tag{19}$$

$$\sum_{i\in\mathcal{C}}\sum_{s\in\mathcal{S}^t} q_{is}^{kt} \leq Q\bar{y}_0^{kt} \qquad k\in\mathcal{K}, t\in\mathcal{T}, \tag{20}$$

$$(2)-(6) \text{ and } (12)-(14),$$

onde $\bar{y}_0^{kt} = 1$ indica que o veículo k é usado no período de tempo t e $\bar{y}_i^{kt} = 1$ indica que o veículo k visita o cliente i no período t. A função objetivo (18) consiste em maximizar o lucro total, dado pela receita total menos os custos de estocagem. As restrições (19) permitem que um veículo faça uma entrega a um cliente em um dado período de tempo apenas se o cliente for de fato visitado pelo veículo nesse período de tempo na solução ($\bar{y}_i^{kt} = 1$). Finalmente, as restrições (20) impõem a capacidade dos veículos. Essa formulação de PL pode ser resolvida usando um *software* de otimização de propósito geral.

Note que visitas vazias podem ocorrer nessa fase, i.e., casos com $\sum_{s\in\mathcal{S}^t} q_{is}^{kt} = 0$ e $\bar{y}_i^{kt} = 1$ para um dado $i\in\mathcal{C}$, $k\in\mathcal{K}$, $t\in\mathcal{T}$. Nesses casos, o cliente é removido da rota e o valor da função objetivo da solução é atualizado.

4. Experimentos computacionais

Nessa seção, são descritos os experimentos computacionais realizados com o método híbrido. Todas as implementações foram feitas em C++ e executadas em uma máquina com processador Intel Xeon X5650 Westmere 2.67 GHz com um único *core* e 36 GB de RAM. O *software IBM CPLEX Optimization Studio v.12.8* foi usado como solver para a formulação apresentada na Seção 2 e a FFM.

Nos experimentos computacionais, foram usados dois conjuntos de instâncias. O primeiro, proposto por Coelho e Laporte [2014] (conjunto CL), é composto por 60 instâncias divididas em 12 subconjuntos com 5 instâncias cada. Além disso, neste trabalho é proposto um segundo conjunto de instâncias, contendo 55 instâncias maiores, divididas em 11 subconjuntos com 5 instâncias cada (conjunto A). Os tamanhos das instâncias para ambos os conjuntos variam em termos do número de clientes (N), idade máxima do produto (S), número de veículos (K) e tamanho do horizonte de planejamento (T). Os intervalos dessas dimensões para ambos os conjuntos estão resumidos na Tabela 1. Observe que as novas instâncias (conjunto A) são maiores em termos da duração do horizonte de planejamento, número de veículos e idade máxima do produto, em vez de em termos do número de clientes.

Parâmetro	Conjunto CL	Conjunto A
N	[10, 50]	[10, 40]
S	[2, 5]	[5, 10]
K	[1, 3]	[2, 8]
T	[3, 10]	[10, 20]

Tabela 1: Dimensões das instâncias utilizadas.

As instâncias foram geradas usando o mesmo procedimento descrito por Coelho e Laporte [2014], exceto r^t e I_{00}^0, dado que os autores não descrevem como os valores desses parâmetros são gerados. Portanto, seguindo a prática da literatura para o PRE [Archetti et al., 2007, 2012] assume-se que o fornecedor tem um suprimento de produtos grande suficiente para atender todos seus clientes em cada período de tempo. Portanto, definiu-se $r^t = 1.5 \sum_{i \in \mathcal{C}} d_i^1,\ \forall t \in \mathcal{T}$ e $I_{00}^0 = r^1$. Vale ressaltar que as instâncias possuem receitas que são não-crescentes com a idade, ou seja, $u_{is} \geq u_{i,s+1},\ \forall s \in \mathcal{S} \backslash \{S\}$, e os custos de estocagem são aleatórios para o fornecedor e para os clientes (usando uma fórmula que inclui um aumento com a idade do produto). Observe que custos de estocagem aleatórios representam o caso mais geral para esses valores. Os custos de viagem correspondem a distâncias Euclidianas arredondadas para o número inteiro mais próximo.

Os resultados obtidos com o método híbrido foram comparados com os da formulação matemática da Seção 2, e são apresentados nas Tabelas 2 e 3. O método híbrido foi executado uma única vez, usando 500 iterações como critério de parada. O valor de `max_perturb` foi definido como $\lceil 0.7 \times N \rceil$. Esse valor foi determinado através de experimentos preliminares empíricos utilizando um subconjunto aleatório das instâncias do problema. O tempo limite imposto para resolver cada instância com a formulação matemática foi de duas horas.

Nas tabelas, a coluna "Diff modelo" mostra a diferença relativa das soluções obtidas com o método híbrido (z^h) para a melhor solução factível encontrada pela formulação matemática (z^f), calculado como $100 \times (z^f - z^h)/z^f$; a coluna "*Gap* otim" mostra o *gap* de otimalidade das soluções obtidas pelo método híbrido, calculado usando a fórmula $100 \times (\bar{z} - z^h)/\bar{z}$, onde $\bar{z}$ é o limitante superior fornecido pelo solver; e a coluna "Tempo" mostra o tempo total (em segundos) requerido pelo método híbrido. Observe que valores negativos para "Diff modelo" indicam que o método híbrido encontrou soluções factíveis melhores do que as fornecidas pela formulação matemática. Cada linha exibe o resultado médio das cinco instâncias do conjunto dado (coluna 1), exceto para a coluna "Diff modelo", cujos valores foram calculados com as instâncias para as quais a formulação foi capaz de encontrar uma solução factível dentro do limite de tempo (duas horas). Além disso, células com um traço ("–") na coluna "Diff modelo" indicam que a formulação não foi capaz de encontrar soluções factíveis para nenhuma das instâncias do respectivo conjunto (coluna 1).

Os resultados para o conjunto de instâncias CL (Tabela 2) mostram que o método proposto é capaz de encontrar soluções de alta qualidade em tempos de execução relativamente pequenos. Especificamente, soluções com um *gap* de otimalidade médio de 2,70% foram obtidas em 5,81 segundos, em média, para todas as instâncias desse conjunto. No mesmo conjunto de instâncias, o tempo médio usando a formulação foi de 3353 segundos. A formulação matemática só encontrou 53 soluções factíveis para o conjunto CL dentro do tempo limite de duas horas, enquanto o método híbrido foi capaz de encontrar soluções factíveis para todas as 60 instâncias deste conjunto. Os resultados também revelam a sensibilidade do método a um aumento no número de períodos (T), principalmente considerando que o esforço requerido na fase de busca local, assim como o tamanho da FFM, depende principalmente T.

Os resultados obtidos com as novas instâncias para o problema (conjunto A, Tabela 3)

https://proceedings.science/p/106697?lang=pt-br

mostram o impacto na resolução do problema ao utilizar instâncias maiores, pois, como esperado, são necessários tempos significativamente maiores para executar as 500 iterações. Além disso, maiores *gaps* de otimalidade são obtidos para as soluções nessas instâncias. Esse fato não reflete necessariamente uma degradação na qualidade das soluções que o método híbrido pode fornecer para essas instâncias, mas pode mostrar a baixa qualidade dos limitantes superiores fornecidos pela formulação. Em média, o método híbrido encontra soluções com valores objetivos 0,35% maiores que as (poucas) soluções encontradas pela formulação. Para esse conjunto de instâncias, a formulação matemática só conseguiu encontrar soluções factíveis para seis instâncias (quatro do conjunto 10-7-2-15 e dois do conjunto 10-10-2-15). Além disso, o tempo médio que o solver usou foi de 7200 segundos, ou seja, a formulação não conseguiu provar a otimalidade para nenhuma das soluções, terminando com um *gap* de otimalidade médio de 12%. Neste conjunto, o método híbrido também foi capaz de encontrar soluções factíveis para todas as instâncias (55 no total).

Conjunto (N-S-K-T)	Diff modelo (%)	*Gap* otim (%)	Tempo (s)
10-2-1-3	0,00	0,00	0,22
10-3-1-6	0,12	0,12	0,35
10-5-1-10	0,64	0,64	0,71
20-2-2-3	0,97	0,97	0,62
20-3-2-6	1,98	3,07	1,31
20-5-2-10	-2,43	4,58	13,08
30-2-2-3	0,41	0,64	1,20
30-3-2-6	0,91	2,29	2,74
30-5-2-10	-1,14	2,89	23,84
40-2-3-3	3,57	6,82	2,06
40-3-3-6	–	5,10	20,22
50-2-3-3	2,65	5,28	3,36
Média	0,77	2,70	5,81

Tabela 2: Resultados com o método híbrido no conjunto de instâncias CL

Conjunto (N-S-K-T)	Diff modelo (%)	*Gap* otim (%)	Tempo (s)
10-7-2-15	-0,14	11,71	12,60
10-10-2-15	1,33	13,92	16,10
10-10-2-20	–	14,90	21,51
20-7-4-15	–	15,87	51,39
20-10-4-15	–	13,34	69,11
20-10-6-15	–	24,59	114,12
30-7-4-15	–	10,09	99,74
30-7-8-15	–	22,84	222,11
30-10-8-15	–	21,07	303,53
40-5-4-10	–	8,93	62,47
40-5-8-10	–	15,18	122,14
Média	0,35	15,68	99,53

Tabela 3: Resultados com o método híbrido no conjunto de instâncias A

5. Conclusões e perspectivas futuras

Neste trabalho, abordou-se uma variante do problema de roteamento de estoques no qual o produto é perecível. Para resolver esse problema, um método híbrido que combina uma metaheurística de busca local iterada com um componente de programação matemática é apresentado.

Além disso, novas instâncias para o problema são introduzidas. Os resultados dos experimentos computacionais mostram que o método de solução proposto é capaz de fornecer soluções de alta qualidade em tempos de execução relativamente curtos para instâncias de pequeno e médio porte. Quando aplicado a instâncias maiores, o método fornece boas soluções factíveis dentro de tempos de execução razoáveis. Uma perspectiva interessante para pesquisas futuras é adaptar o método híbrido para resolver extensões mais ricas do problema, como o caso de vários produtos perecíveis, ou ainda a versão estocástica do problema.

Agradecimentos

Os autores agradecem o apoio financeiro da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), processos 2017/06664-9, 2017/13739-5, 2016/23366-9 e 2016/01860-1; do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

Referências

Adulyasak, Y., Cordeau, J.-F., e Jans, R. (2014). Formulations and branch-and-cut algorithms for multivehicle production and inventory routing problems. *INFORMS Journal on Computing*, 26 (1):103–120.

Alvarez, A., Munari, P., e Morabito, R. (2018). Iterated local search and simulated annealing algorithms for the inventory routing problem. *International Transactions in Operational Research*, 25:1785–1809.

Archetti, C., Bertazzi, L., Laporte, G., e Speranza, M. G. (2007). A branch-and-cut algorithm for a vendor-managed inventory-routing problem. *Transportation Science*, 41(3):382–391.

Archetti, C., Bertazzi, L., Hertz, A., e Speranza, M. G. (2012). A hybrid heuristic for an inventory routing problem. *INFORMS Journal on Computing*, 24(1):101–116. ISSN 10919856.

Azadeh, A., Elahi, S., Farahani, M. H., e Nasirian, B. (2017). A genetic algorithm-Taguchi based approach to inventory routing problem of a single perishable product with transshipment. *Computers & Industrial Engineering*, 104:124–133.

Coelho, L. e Laporte, G. (2014). Optimal joint replenishment, delivery and inventory management policies for perishable products. *Computers & Operations Research*, 47:42 – 52.

Crama, Y., Rezaei, M., Savelsbergh, M., e Woensel, T. V. (2018). Stochastic inventory routing for perishable products. *Transportation Science*, 52(3):526–546.

Diabat, A., Abdallah, T., e Le, T. (2016). A hybrid tabu search based heuristic for the periodic distribution inventory problem with perishable goods. *Annals of Operations Research*, 242(2): 373–398.

Federgruen, A., Prastacos, G., e Zipkin, P. (1986). An allocation and distribution model for perishable products. *Operations Research*, 34(1):75–82. ISSN 0030-364X.

Hemmelmayr, V., Doerner, K., Hartl, R., e Savelsbergh, M. (2009). Delivery strategies for blood products supplies. *OR Spectrum*, 31(4):707–725. ISSN 01716468.

Le, T., Diabat, A., Richard, J.-P., e Yih, Y. (2013). A column generation-based heuristic algorithm for an inventory routing problem with perishable goods. *Optimization Letters*, 7(7):1481–1502. ISSN 1862-4472.

Lourenço, H., Martin, O., e Stützle, T. (2010). Iterated local search: Framework and applications. In *Handbook of Metaheuristics*, volume 146, p. 363–397.

Mirzaei, S. e Seifi, A. (2015). Considering lost sale in inventory routing problems for perishable goods. *Computers & Industrial Engineering*, 87:213–227. ISSN 03608352.

Mladenović, N. e Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research*, 24:1097–1100.

Rohmer, S., Claassen, G., e Laporte, G. (2019). A two-echelon inventory routing problem for perishable products. *Computers & Operations Research*, 107:156–172.