São José dos Campos - SP
LV SBPO
LV Simpósio Brasileiro de Pesquisa Operacional
São José dos Campos, SP - 6 a 9 de novembro de 2023

UM MODELO DE PROGRAMAÇÃO LINEAR INTEIRA MISTA PARA UM PROBLEMA DE TRANSFERÊNCIA E TRANSPORTE DE VALORES: O CASO DE UM BANCO PERUANO

Manuel Valois Castillo Quispe, Raphael Kramer
Departamento de Engenharia de Produção, Universidade Federal de Pernambuco
Av. da Arquitetura, s/n, Cidade Universitária, 50740-550, Recife–PE
raphael.kramer@ufpe.br, manuel.castillo@ufpe.br

Walton Pereira Coutinho
Núcleo de Tecnologia da Informação, Universidade Federal de Pernambuco
Av. Campina Grande, s/n - Km 59 - Nova Caruaru, 55014-900, Caruaru–PE.
walton.coutinho@ufpe.br

Luciano Carlos Azevedo da Costa
Departamento de Engenharia de Produção, Universidade Federal da Paraíba
Campus Universitário de João Pessoa - Campus I, 58051-970, João Pessoa–PB
luciano.costa@academico.ufpb.br

RESUMO

As operações logísticas de transferência física de cédulas e moedas (*cash-in-transit*) costumam envolver custos e riscos elevados. No entanto, para atender adequadamente à demanda, os bancos precisam administrar seus custos operacionais, de armazenamento e transporte. Neste trabalho, analisamos a rede de distribuição de um banco espanhol que opera no Peru. Essa rede de distribuição é composta por agências locais, varejistas e caixas eletrônicos. Uma formulação de problema de programação inteira mista multi-período e multi-*commodity* é proposta para resolver este problema. Além disso, para resolver instâncias do mundo real maiores e mais difíceis, duas heurísticas baseadas em horizonte rolante são propostas, a saber, *relax-and-fix* e *fix-and-optimize*. Experimentos computacionais preliminares mostram que as abordagens propostas são capazes de encontrar soluções eficientes para as instâncias propostas.

PALAVRAS CHAVE. Cash-in-transit. Logística. Programação Inteira Mista. Tópicos: PM – Programação Matemática. GF – Gestão Financeira.

ABSTRACT

Logistic operations regarding the physical transfer of banknotes and coins (cash-in-transit) usually involve high costs and risks. However, in order to properly meet demand requirements, banks need to manage their operational, storage and transportation costs. In this work, we analyze the distribution network of a Spanish bank operating in Peru. This distribution network is composed of local branches, retailers, and ATMs. A multi-period and multi-commodity mixed-integer programming problem formulation is proposed to tackle this problem. In addition, to solve larger and harder real-world-based instances, two heuristics based on rolling horizon are proposed, namely, relax-and-fix and fix-and-optimize. Preliminary computational experiments show that the proposed approaches are capable of efficiently finding solutions to the proposed instances.

KEYWORDS. Cash-in-transit. Logistics. Mixed-integer programming.
Paper topics: PM - Mathematical Programming. GF - Financial management.

DOI: 10.59254/sbpo-2023-175184
http://live-proceedings.pantheonsite.io/p/175184

1. Introdução

As operações logísticas de transferência física de cédulas e moedas (*cash-in-transit*) costumam envolver custos e riscos elevados. Os bancos precisam lidar com diversos custos, como: custos de operação para atender as demandas dos clientes; custos de estocagem; custos operacionais e de transporte, relacionados com a movimentação do dinheiro em um centro de distribuição; entre outras [Zhou et al., 2020]. O bom gerenciamento das operações logísticas relacionadas com a armazenagem e distribuição de cédulas e moedas apresentam grande importância, pois elas impactam tanto no atendimento das demandas dos clientes quanto na redução de gastos que podem ser gerados para um banco. A correta operacionalização do processo pode reduzir consideravelmente os gastos operacionais do banco, bem como manter o cliente satisfeito [van Anholt, 2014].

Na literatura, alguns trabalhos já se interessaram pela temática em questão. Diao et al. [2016] propuseram um framework baseado modelo de programação linear inteira para representar o problema de distribuição de cédulas em caixas eletrônicos. O framework proposto possui dois níveis de decisão. O primeiro nível visa a minimização dos custos de oportunidade (cédulas ociosas no caixa) e dos custos de recarga em dinheiro (o custo operacional para viagens de recarga de caixas eletrônicos). Por sua vez, o segundo nível fornece uma relação de estoque mínimo que representa o trade-off necessário em ter excesso de caixa, que ao ser passado para o primeiro nível permite a definição das quantidades ideias recarga e os cronogramas de entrega. Os resultados obtidos demostraram a eficácia da abordagem proposta.

Batı e Gözüpek [2019] estudaram o problema de distribuição de cédulas em caixas-eletrônicos de nova geração. Nestes caixas, as notas depositas e sacadas são armazenadas no mesmo compartimento. Isso faz com os equipamentos precisem ser reabastecidos com menos frequência. Para resolver o problema em estudo, os autores propõem um modelo de programação inteira que decide conjuntamente quando visitar um caixa eletrônico, quanto dinheiro entregar em qual caixa eletrônico e qual caminho deve ser percorrido para a distribuição de dinheiro nos caixas eletrônicos. O modelo também decide quais caixas eletrônicos devem ser substituídos por caixas eletrônicos de nova geração. Devido à complexidade do modelo proposto, os autores propõem um algoritmo heurístico que foi capaz de gerar soluções de qualidade em tempos satisfatórios, o que torna o método adequado para ser implementado na prática. Özer [2019] também analisa o problema de reabastecimento de caixas eletrônicos. O autor propõe um modelo de programação matemática que visa reduzir o número de vezes e os períodos em que os caixas eletrônicos ficam desabastecidos. Para resolver o problema, o autor propõe um algoritmo de programação dinâmica.

O trabalho em questão analisa o caso de um banco espanhol que precisa distribuir dinheiro para diversos centros de distribuição em diferentes províncias do Peru. Esses centros de distribuição podem ser representado por como agências, *retailers* e/ou caixas eletrônicos. Para cada um deles, o banco deve decidir a quantidade de cédulas a ser fornecida para satisfazer a demanda do cliente, garantindo que os custos operacionais serão minimizados. Para resolver o problema propõe-se um modelo multi-período e *multicommodity* de programação linear inteira para minimizar os custos operacionais. Uma descrição detalhada do problema é apresentada na Seção 2. Devido ao grande número de restrições, o modelo é apenas capaz de lidar com problemas de pequeno porte. Por esse motivo, são propostas duas heurísticas de horizonte deslizante, as quais são executadas sequencialmente: (i) *relax-and-fix* (R&F) e a (ii) *fix-and-optimize* (F&O).

O restante do artigo está estruturado como segue: Na Seção 2, apresenta-se uma descrição detalhada do problema em estudo. Na Seção 3, apresenta-se o modelo de programação matemática utilizado para representar o problema. Na Seção 4, são descritas as heurísticas de horizonte deslizante propostas para resolver o problema. Na Seção 5 são apresentados experimentos computa-

DOI: 10.59254/sbpo-2023-175184
http://live-proceedings.pantheonsite.io/p/175184

cionais a fim de se demonstrar a eficiência da dos métodos propostos. Por fim, na Seção 6, são conclusões são apresentadas.

2. Descrição do Problema

O problema objeto de estudo deste artigo é derivado de um caso real enfrentado por um banco espanhol com atuação no Peru, referente ao abastecimento de dinheiro nos diversos pontos de distribuição (e.g., caixas eletrônicos, agências bancárias, lojas varejistas). Diariamente, o banco deve decidir a quantidade de cada tipo de nota a ser disponibilizada nos pontos de distribuição, visando o atendimento da demanda (prevista) de seus clientes.

O abastecimento de dinheiro é operacionalizado por meio de empresas terceirizadas de transporte e segurança (armazenamento) de valores e por meio de transferências (saques e depósitos) junto ao *Banco Central de Reserva* (BCR). O transporte das notas pode ser realizado nos modais aéreo e terrestre. Já o armazenamento é feito nos cofres das empresas terceirizadas (e/ou do BCR). O objetivo do problema consiste em minimizar os custos de transporte e armazenamento de dinheiro para atender a demanda agregada de cada cidade em um dado horizonte de planejamento, os quais dependem do operador considerado (empresas terceirizadas ou BCR), do modal utilizado, da quantidade de notas e do valor armazenado ou transportado.

Formalmente, o problema pode ser representado por um grafo $G = (N, A)$, onde N é o conjunto de nós e A é o conjunto de arcos. Cada nó representa um ponto de armazenamento de dinheiro, estando associado a uma cidade e a uma empresa terceirizada (ou BCR). O conjunto de arcos representa as conexões entre nós de cidades distintas, indicando a possibilidade de transporte de valores em qualquer tipo de modal. Seja C o conjunto de cidades e $N_c \subset N$ o conjunto de nós associados a um cofre de uma empresa terceirizada na cidade $c \in C$. Assim, o conjunto de arcos pode ser definido como $A = \{(i, j) : c \in C,\ i \in N_c,\ j \in N \setminus N_c\}$. Tomando como exemplo o segundo grafo da Figura 1 (i.e., aquele à direita), temos $C = \{1, 2, 3, 4, 5\}$, $N = \{1, 2, 3, 4, 5, 6, 7\}$, $N_1 = \{1, 2\}$, $N_2 = \{3, 4\}$, $N_3 = \emptyset$, $N_4 = \{5\}$, $N_5 = \{6, 7\}$, e $A = \{(1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (2, 3), (2, 4), (2, 5), (2, 6), (2, 7), (3, 5), (3, 6), (3, 7), (4, 5), (4, 6), (4, 7), (5, 6), (5, 7)\}$.

Seja D o conjunto de tipos de notas (e.g., 10, 20, 50, 100 e 200 soles, reais, etc.), T o conjunto de períodos (e.g., dias), representando o horizonte de planejamento, e M o conjunto de modais de transporte. Tomando como base o grafo G definido anteriormente, cada nó $i \in N$ possui um estoque inicial I^0_{ik} de notas do tipo $k \in D$ e um custo unitário de armazenamento h_{ik} para cada nota do tipo $k \in D$. O valor de uma nota do tipo $k \in D$ é denotado por v_k. Cada cidade $c \in C$ possui uma demanda q_{ckt} por notas do tipo $k \in D$ no período $t \in T$, e cada arco (i, j) possui um custo de transporte c_{ijm} (por veículo) dependente da distância entre os nós $i \in N$ e $j \in N$ e do modal $m \in M$ utilizado. Assume-se que todos os veículos de um mesmo modal são idênticos, onde Q_m é a capacidade de um veículo do modal $m \in M$, em quantidade de notas.

Outro componente de custo depende do valor transportado. Definindo F_m como sendo o conjunto de faixas de valores associados ao modal $m \in M$, ordenadas de maneira crescente, denota-se por C_{mf} o custo de transporte no modal $m \in M$ na faixa de valor $f \in F_m$, com $C_{mf} > C_{mf-1}$. Uma faixa de valor $f \in F_m$ é representada por um limite inferior LB_{mf} e um limite superior K_{mf}. Por fim, consideram-se que os custos de sacar ou depositar dinheiro no BCR também são definidos por faixas de valores. Sejam F^S e F^D os conjuntos de faixas de valores para saque ou depósito no BCR, respectivamente. Assim, depositar (ou sacar) no (ou do) BCR um valor pertence ao intervalo $[LB^D_f, K^D_f]$ (ou $[LB^S_f, K^S_f]$) resulta em um custo C^D_f (ou C^S_f), por transação. Assume-se que valores depositados em cofres do BCR não acarretam custos de armazenamento para a empresa. Dessa forma, podemos interpretar as operações de depósito ou saque no/do BCR como sendo a

DOI: 10.59254/sbpo-2023-175184
http://live-proceedings.pantheonsite.io/p/175184

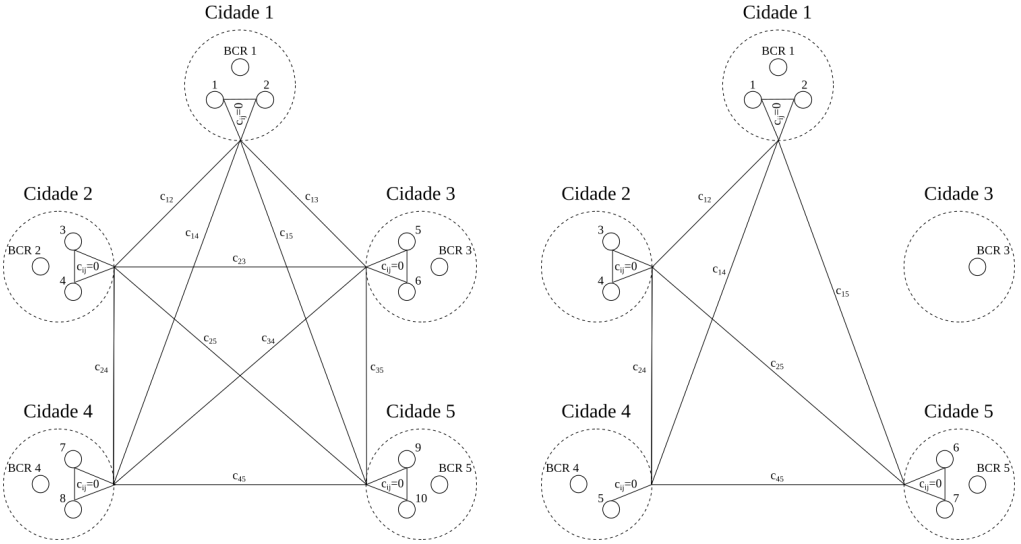


Figura 1: Exemplo de dois grafos envolvendo 5 cidades. À esquerda, um grafo completo em que todas as cidades possuem um cofre do BCR e dois cofres de empresas terceirizadas. À direita, um grafo não-completo onde nem todas as cidades possuem cofres de empresas terceirizadas e/ou do BCR.

entrada ou saída de dinheiro no/do sistema, respectivamente.

Considerando as definições apresentadas, o problema consiste em decidir para cada período $t \in T$ e para cada tipo de nota $k \in D$, (i) quanto transportar de um cofre (ou cidade) para outro(a), (ii) quanto armazenar em cada cofre (ou cidade), (iii) quanto sacar do BCR e (iv) quanto depositar no BCR, de modo a atender a demanda diária de cada cidade, visando a minimização do custo total que engloba operações de transporte e armazenamento de valores e transações junto ao BCR.

3. Modelo Matemático

Fazendo uso das definições apresentadas anteriormente e das variáveis de decisão listadas abaixo, o problema pode então ser modelado como um problema de programação linear inteira mista, conforme (1)-(17).

- x_{ijkmt} – quantidade de notas do tipo $k \in D$ transportadas de $i \in N$ para $j \in N$ pelo modal $m \in M$ no período $t \in T$;
- I_{ikt} – quantidade de notas do tipo $k \in D$ estocadas em $i \in N$ no período $t \in T$;
- y^S_{ctf} – se a faixa de valor $f \in F^D$ é usada para fazer saque do BCR da cidade $c \in C$ no período $t \in T$;
- $y^D_{ctf} = \{0, 1\}$ – se a faixa de valor $f \in F^D$ é usada para depositar no BCR da cidade $c \in C$ no período $t \in T$;
- $y_{ijmtf} = \{0, 1\}$ – se a faixa de valor $f \in F_m$ é usada para transportar valor de $i \in N$ para $j \in N$ com o modal $m \in M$ no período $t \in T$;
- s_{ckt} – quantidade de notas sacadas do BCR da cidade $c \in C$ no período $t \in T$;
- d_{ckt} – quantidade de notas depositadas no BCR da cidade $c \in C$ no período $t \in T$;
- $w_{ijmt} \in \mathbb{Z}_{\geq 0}$ – quantidade de veículos do modal $m \in M$ utilizados de $i \in N$ para $j \in N$ no período $t \in T$.

$$\text{Min} \sum_{i \in N} \sum_{j \in N} \sum_{m \in M} \sum_{t \in T} c_{ijm} w_{ijmt} +$$

DOI: 10.59254/sbpo-2023-175184
http://live-proceedings.pantheonsite.io/p/175184

$$\begin{aligned}
&\sum_{i\in N}\sum_{j\in N}\sum_{m\in M}\sum_{t\in T}\sum_{f\in F_m} C_{mf}y_{ijmtf} + \\
&\sum_{i\in N}\sum_{k\in D}\sum_{t\in T} h_{ik}I_{ikt} + \\
&\sum_{c\in C}\sum_{t\in T}\sum_{f\in F^S} C^S_f y^S_{ctf} + \\
&\sum_{c\in C}\sum_{t\in T}\sum_{f\in F^D} C^D_f y^D_{ctf} \qquad (1)\\
&\text{sujeito a } \sum_{j\in N_c} I_{jkt-1} + \\
&\Big(\sum_{i\in N}\sum_{j\in N}\sum_{m\in M} x_{ijkmt} - \sum_{j\in N_c}\sum_{i\in N}\sum_{m\in M} x_{jikmt}\Big) + \\
&s_{ckt} - d_{ckt} - \sum_{j\in N_c} I_{jkt} = q_{ckt}, \qquad c\in C, k\in D, t\in T\setminus\{0\}, \quad (2)\\
&\sum_{f\in F^S} LB^S_f y^S_{ctf} \le \sum_{k\in D} v_k s_{ckt} \le \sum_{f\in F^S} K^S_f y^S_{ctf}, \qquad c\in C, t\in T, \quad (3)\\
&\sum_{f\in F^S} y^S_{ctf} \le 1, \qquad c\in C, t\in T, \quad (4)\\
&\sum_{f\in F^D} LB^D_f y^D_{ctf} \le \sum_{k\in D} v_k d_{ckt} \le \sum_{f\in F^D} K^D_f y^D_{ctf}, \qquad c\in C, t\in T, \quad (5)\\
&\sum_{f\in F^S} y^D_{ctf} \le 1, \qquad c\in C, t\in T, \quad (6)\\
&\sum_{f\in F_m} LB_{mf}y_{ijmtf} \le \sum_{k\in D} v_k x_{ijkmt} \le \sum_{f\in F_m} K_{mf}y_{ijmtf}, \qquad i,j\in N, m\in M, t\in T, \quad (7)\\
&\sum_{f\in F_m} y_{ijmtf} \le 1, \qquad i,j\in N, m\in M, t\in T, \quad (8)\\
&w_{ijmt} \ge \sum_{k\in D} x_{ijkmt}/Q_m, \qquad i,j\in N, m\in M, t\in T, \quad (9)\\
&I_{jk0} = I^0_{jk}, \qquad j\in N, k\in D, \quad (10)\\
&x_{ijkmt}\in\{0,1\}, \qquad i,j\in N, k\in D, m\in M, t\in T, \quad (11)\\
&I_{kt}\in\{0,1\}, \qquad i\in N, k\in D, t\in T, \quad (12)\\
&y^S_{ctf}\in\{0,1\}, \qquad c\in C, t\in T, f\in F^S, \quad (13)\\
&y^D_{ctf}\in\{0,1\}, \qquad c\in C, t\in T, f\in F^D, \quad (14)\\
&y_{ijmtf}\in\{0,1\}, \qquad i,j\in N, m\in M, t\in T, f\in F_m \quad (15)\\
&s_{ckt}, d_{ckt}\in\{0,1\}, \qquad c\in C, k\in D, t\in T, \quad (16)\\
&w_{ijmt}\in\mathbb{Z}_{\ge 0}, \qquad i,j\in N, m\in M, t\in T. \quad (17)
\end{aligned}$$

A Função Objetivo (1) minimiza o custo total de transporte, armazenamento e transações junto ao BCR, sendo composta por 5 termos. O primeiro e segundo termos calculam os custos de transporte, os quais dependem da distância percorrida, da quantidade de veículos utilizados e

DOI: 10.59254/sbpo-2023-175184
http://live-proceedings.pantheonsite.io/p/175184

do valor transportado. O terceiro termo calcula o custo de armazenamento, enquanto o quarto e o quinto termos calculam os custos de efetuar saques e depósitos junto ao BCR, respectivamente. O conjunto de Restrições (2) garante que as demandas das cidades sejam satisfeitas em todos os períodos, isto é, a quantidade estocada no final do período anterior somada às quantidades que chegam via transporte e/ou saque do BCR e subtraída às quantidades que saem via transporte e/ou depósito no BCR e que ficam armazenadas para o(s) período(s) seguinte(s), deve ser igual à q_{ckt}, $\forall c \in C, k \in D, t \in T$. As Restrições (3) e (4) asseguram que, caso haja saque do BCR, apenas uma faixa de valor seja utilizada e que essa seja condizente com o valor sacado. As Restrições (5) e (6) são similares às anteriores, porém para depósitos no BCR. As Restrições (7) e (8) também são similares às anteriores, porém para o transporte de valores. O conjunto de Restrições (9) define a quantidade mínima de veículos para transportar valores entre nós (cofres), para cada modal e período. As Restrições (10) estabelecem o nível de estoque inicial de cada tipo de nota em cada cofre. Por fim, os domínios das variáveis são definidos em (11)-(17).

4. Heurísticas baseadas em horizonte deslizante

Dado o número bastante elevado de variáveis e restrições do modelo matemático proposto na Seção 3, resolvê-lo na otimalidade demanda bastante esforço computacional, até mesmo para instâncias de dimensões pequenas. Não apenas encontrar uma solução ótima, mas encontrar uma solução viável de boa qualidade demanda bastante tempo de processamento computacional.

Considerando a natureza periódica do problema e visando encontrar soluções viáveis de boa qualidade em um tempo relativamente curto, optou-se pela utilização de duas heurísticas de horizonte deslizante, executadas sequenciadamente: a (i) *relax-and-fix* (R&F) e a (ii) *fix-and-optimize* (F&O). Ambas as heurísticas (R&F e F&O) consistem na decomposição do modelo (1)-(17) em subproblemas que são consideravelmente mais fáceis de serem resolvidos [Apolinário, 2020].

Na R&F, algumas (ou todas as) variáveis de decisão – dentre aquelas definidas originalmente no domínio dos inteiros – são decompostas em três subconjuntos: (i) um subconjunto de variáveis têm suas restrições de integralidade relaxadas, (ii) outro subconjunto tem seus valores fixados e, por fim, (iii) o restante das variáveis são mantidas conforme definidas no modelo original. Por exemplo, seja $v \in \mathbb{Z}^n$ um vetor de n variáveis de decisão pertencentes ao conjunto dos inteiros. Na decomposição, poderíamos ter $v_t = valor_fixo_t, \forall t \in \{1, \ldots, \alpha - 1\}$, $v_t \in \mathbb{Z}, \forall t \in \{\alpha, \ldots, \beta\}$, e $v_t \in \mathbb{R}, \forall t \in \{\beta + 1, \ldots, |T|\}$, conforme ilustrado na Figura 2. De maneira similar, na heurística F&O um subconjunto das variáveis de decisão têm seus valores fixados e o restante das variáveis são mantidas conforme definidas no modelo original. No exemplo anterior, poderíamos ter $v_t = valor_fixo_t, \forall t \in \{1, \ldots, \alpha - 1, \beta + 1, \ldots, |T|\}$ e $v_t \in \mathbb{Z}, \forall t \in \{\alpha, \ldots, \beta\}$, conforme ilustrado na Figura 3. Os subproblemas são atualizados e resolvidos iterativamente até que se obtenha uma solução viável (no caso da heurística R&F) e/ou até que um critério de parada seja satisfeito. A cada iteração os parâmetros α e β vão sendo incrementados (por um valor k'), até que β seja igual a $|T|$. Na prática, a heurística R&F pode ser vista como uma *heurística construtiva*, enquanto a F&O pode ser vista como uma *heurística de refinamento*. Os Algoritmos 1 e 2 apresentam a estrutura lógica das heurísticas R&F e F&O, respectivamente.

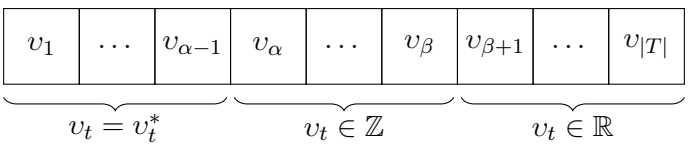


Figura 2: Redefinição do domínio das variáveis na heurística *relax-and-fix*.

DOI: 10.59254/sbpo-2023-175184
http://live-proceedings.pantheonsite.io/p/175184

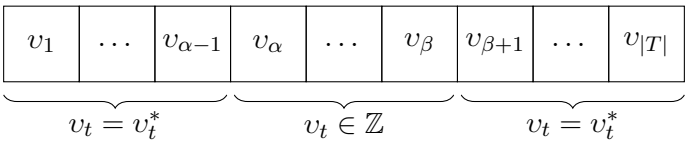


Figura 3: Redefinição do domínio das variáveis na heurística *fix-and-optimize*.

Algoritmo 1 Heurística *Relax-and-Fix*

1: **Input:** k, v
2: $k' \leftarrow k - \lfloor k/2 \rfloor$
3: $\alpha \leftarrow 1,\ \beta \leftarrow \alpha + k - 1$
4: $v^* \leftarrow \emptyset$
5: **enquanto** *true* **faça**
6: $\quad modeloRF \leftarrow \texttt{CriaModeloRF}(v, v^*, \alpha, \beta)$
7: $\quad S, v^* \leftarrow \texttt{Solve}(modeloRF)$ ▷ Onde, $v^* \subseteq S$
8: $\quad$ **se** $\beta = |T|$ **então**
9: $\quad\quad$ **break**
10: $\quad \alpha \leftarrow \alpha + k', \beta \leftarrow min\{\alpha + k - 1, |T|\}$
11: **retorne** S;

A heurística R&F, representada pelo Algoritmo 1, recebe como dados de entrada o valor do parâmetro k e o subconjunto de variáveis que terão seus domínios alterados ao longo da execução do algoritmo. Com base no valor de k, calcula-se o valor do passo k e os valores iniciais dos parâmetros α e β, que irão definir os subconjuntos de variáveis que terão seus domínios modificados (ver Figura 2). Vale frisar que na primeira iteração nenhuma variável possui valor fixado, enquanto que na última iteração nenhuma variável possui a restrição de integralidade relaxada. Uma vez definidos os valores de α e β, constrói-se o modelo matemático relaxado e o resolve (Linhas 6 e 7). Em seguida, atualiza-se o intervalo $[\alpha, \beta]$, deslocando-o k' unidades para a direita (Linha 10). O algoritmo termina sua execução após resolver o modelo com $\beta = |T|$, retornando uma solução viável S.

DOI: 10.59254/sbpo-2023-175184
http://live-proceedings.pantheonsite.io/p/175184

Algoritmo 2 Heurística *Fix-and-Optimize*

1: **Input:** k, υ, S
2: $k' \leftarrow k - \lfloor k/2 \rfloor$
3: $\alpha \leftarrow 1;\ \beta \leftarrow \alpha + k - 1$
4: $S^*, \upsilon^* \leftarrow S$ ▷ Onde, $\upsilon^* \subseteq S$
5: $improved \leftarrow true$
6: **enquanto** $improved = true$ **faça**
7: $improved \leftarrow false$
8: **enquanto** $true$ **faça**
9: $modeloFO \leftarrow$ `CriaModeloFO`$(\upsilon, \upsilon^*, \alpha, \beta)$
10: $S, \upsilon^* \leftarrow$ `Solve`$(modeloFO)$ ▷ Onde, $\upsilon^* \subseteq S$
11: **se** $f(S) < f(S^*)$ **então**
12: $S^* \leftarrow S$
13: $improved \leftarrow true$
14: **se** $\beta = |T|$ **então**
15: **break**
16: $\alpha \leftarrow \alpha + k', \beta \leftarrow min\{\alpha + k - 1, |T|\}$
17: **se** Melhora Percentual $\leq \Delta_1$ **e** $k < |T| - 1$ **então**
18: $k \leftarrow k + 1;\ \ k' \leftarrow k - \lfloor k/2 \rfloor$
19: **se** Melhora Percentual $< \Delta_2$ **e** $k = |T| - 1$ **então**
20: **break**
21: **retorne** S^*;

Já a heurística F&O, representada no Algoritmo 2, recebe os mesmos dados de entrada da heurística R&F acrescido de uma solução viável S. Primeiramente, inicializam-se os parâmetros k', α e β e faz-se uma cópia da solução S em S^*, definida como sendo a melhor solução conhecida (solução incumbente). A partir da solução S, obtém-se os valores inicias para as variáveis $\upsilon_t, \forall t \in \{1, \ldots, \alpha - 1, \beta + 1, \ldots, |T|\}$ a terem seus valores fixados no modelo matemático. Assim como no algoritmo anterior, vale frisar que na primeira iteração (do loop *while* das Linhas 8-16) apenas as variáveis υ_t com $t \in \{\beta + 1, \ldots, |T|\}$ terão seus valores fixados, enquanto que na última iteração (do loop *while* das Linhas 8-16) apenas as variáveis υ_t com $t \in \{1, \ldots, \alpha - 1\}$ terão seus valores fixados. No *loop* principal do algoritmo (Linhas 8-16), cria-se o modelo matemático definindo o domínio das variáveis como apresentado anteriormente (ver Figura 2) e o resolve (Linhas 9 e 10). Após a resolução do modelo, verifica-se se a solução obtida é melhor que a solução incumbente S^*, atualizando-a em caso positivo, e calculam-se os novos valores dos parâmetros α e β. O *loop* principal encerra após a resolução do modelo com com $\beta = |T|$. Se a melhora percentual obtida após a execução do *loop* principal for maior que Δ_1, este deve ser reinicializado. Caso contrário, se melhora percentual for menor ou igual a Δ_1 e k for menor que $|T| - 1$, então incrementa-se o valor de k em uma unidade, atualiza-se o valor de k' (Linhas 17-18), e reinicializa-se o *loop* principal. O algoritmo F&O encerra sua execução quando a melhora percentual obtida após a execução do *loop* principal for inferior a Δ_2 e k for igual a $|T| - 1$. Ao término, retorna-se a melhor solução encontrada S^*.

Como forma de reduzir o esforço computacional, ambos os algoritmos podem fazer uso de tempos limite para resolução dos subproblemas. No Algoritmo 1, em particular, devemos tomar o cuidado de definir um valor que permita a obtenção de uma solução viável. No caso do Algoritmo

DOI: 10.59254/sbpo-2023-175184
http://live-proceedings.pantheonsite.io/p/175184

2, outro tempo limite pode ser imposto para a execução do algoritmo em si, pois, dado que o mesmo recebe uma solução viável como *input*, temos a garantia de que uma solução viável será retornada a qualquer momento em que o algoritmo tiver sua execução interrompida.

5. Experimentos computacionais

5.1. Ambiente computacional

As implementações do modelo descrito na Seção 3 e dos algoritmos apresentados na Seção 4 foram feitas em linguagem de programação Julia 1.6.4, sendo o JuMP a biblioteca utilizada para implementação dos modelos matemáticos. Os experimentos computacionais foram realizados em um computador com processador Intel Core i5-9300H (2.40GHz) com 16 GB de memória RAM, e sistema operacional Linux Ubuntu 20.04 LTS. O *solver* utilizado para resolução dos modelos matemáticos foi o Gurobi 9.0.2.

5.2. Instâncias e parametrização

Foram consideradas 5 instâncias, estas geradas tomando como base dados reais coletados junto à empresa objeto desse estudo, no mês de junho de 2022. Tais instâncias envolvem 24 cidades, 30 cofres, 5 tipos de notas (10, 20, 50, 100 e 200 *sols peruanos*), duas empresas terceirizadas para transporte e armazenamento de valores e $|T| \in \{3, 4, 5, 10, 20\}$ dias.

Os dados armazenados referentes aos estoques iniciais e às demandas em quantidade de notas não foram coletados por completo, de tal forma que foi necessário fazer uma estimativa para tais valores. Tal estimativa foi realizada com base no valor monetário equivalente às demandas em quantidade de notas – valor esse inferior à demanda real (conhecida), em valor monetário – e na proporção de cada tipo de nota em relação a esse valor monetário. A partir dessas proporções e da demanda (real) em valor monetário, foram obtidas as quantidades demandadas por tipo de nota. Por exemplo, seja DV_c a demanda da cidade c em valor monetário e DQ_{ck} a demanda da cidade c por notas do tipo k, em quantidade. Como a demanda por quantidade de notas não foi armazenada (coletada) por completo, temos que $\sum_{k \in D} DQ_{ck} \cdot v_k < DV_c$. Sendo $p_{ck} = \frac{DQ_{ck} \cdot v_k}{\sum_{k' \in D} DQ_{ck'} \cdot v_k}$ a proporção estimada do valor de notas do tipo k em relação à demanda (real) em valor monetário, a demanda por notas do tipo k, em quantidade, foi estimada como sendo $q_{ck} = \left\lceil \frac{p_{ck} DV_c}{v_k} \right\rceil$.

Foram consideradas três faixas de valores (monetários) para transporte tanto no modal terrestre (1) quanto no aéreo (2), i.e., $|F_m| = 3, m \in \{1, 2\}$. Com base no limite superior da faixa mais elevada (K_{m3}) e no valor da nota menos valiosa (v_1), estimou-se a capacidade dos veículos em quantidade de notas como sendo $Q_m = \left\lfloor \frac{K_{m3}}{v_1} \right\rfloor$. Para depósito e saque (no) do BCR, foram consideradas quatro faixas de valores. As faixas de valores e seus respectivos custos são apresentadas na Tabela 1, assim como as capacidades de transporte.

Os custos c_{ij1} para cada arco (i, j) para o modal terrestre ($m = 1$) foram estimados tomando como base a distância (e.g., em metros) entre as cidades, o consumo (e.g., em litro/metro) e o preço (e.g., em *sol*/litro) do combustível. Já os custos para o modal aéreo ($m = 2$) foram estimados tomando como referência os preços das passagens. Nesse segundo caso, em alguns trechos (arcos) não é permitido (por política interna) o transporte aéreo ($c_{ij2} = \infty$) e onde são permitidos, por simplificação, os custos foram fixados em 2000 *sols*. Por fim, os custos de estoque por dia, para cada tipo de nota, foram calculados tomando como base o valor da nota e a tarifa de 1,27 *sol* para cada 100.000 *sols*.

Os valores dos parâmetros considerados na resolução de cada instância são apresentados na Tabela 2. "TL (s)" se refere ao tempo limite, em segundos, para resolução do modelo matemático (1)-(17) e para execução das heurísticas R&F e F&O. "TL R&F (s)" e "TL F&O (s)" se referem aos tempos limites, em segundos, para resolução dos modelos matemáticos nas heurísticas R&F e F&O, respectivamente. A coluna "k", por sua vez, indica os valores adotados para o parâmetro k,

DOI: 10.59254/sbpo-2023-175184
http://live-proceedings.pantheonsite.io/p/175184

Tabela 1: Capacidade por modal e faixas de valores e custos para transporte e saque/depósito via BCR

Forma de transferência de valores	Capacidade (qtd.)	Faixa	Limites (monet.)		Custo
			Inferior	Superior	
Transporte terrestre	55000	1	1000	119000	120
		2	119001	215000	200
		3	215001	550000	500
Transporte aéreo	110000	1	2000	119000	390
		2	119001	215000	550
		3	215001	250000	1150
		4	250001	550000	1350
		5	550001	563000	1750
		6	563001	1100000	2650
Saque/Depósito BCR	-	1	1000	2000	120
		2	2001	50000	250
		3	50001	802000	400
		4	802001	4010000	866

que define o tamanho (inicial) do conjunto $\{\alpha, \ldots, \beta\}$, e, por fim, a coluna "$\upsilon$" indica as variáveis que terão seus domínios alterados durante a execução das heurísticas R&F e F&O.

Tabela 2: Parâmetros

Instância		Parâmetros				
#	$\|T\|$	TL (s)	TL R&F (s)	TL F&O (s)	k	υ
1	4	3600	20	20	2	x, I, y, w
2	5	3600	20	20	3	x, I, y, w
3	6	3600	30	20	3	x, I, y, w
4	11	3600	60	40	4	x, I, y, w
5	21	3600	120	40	5	x, I, y, w

5.3. Resultados preliminares

Um resumo dos resultados obtidos é apresentado na Tabela 3. As duas primeiras colunas referenciam as instâncias, onde $|T|$ indica o número de períodos considerados em cada instância. As demais colunas reportam os resultados obtidos pelo modelo matemático (1)-(17) e pelas heurísticas de horizonte deslizante (R&F e F&O) apresentadas na Seção 4. As colunas *Custo*, *CPU T (s)* e *Gap-LB(%)* reportam, para cada método, o valor da melhor solução viável encontrada, o tempo computacional utilizado e o *gap* em relação ao *lower bound* retornado com a resolução modelo (1)-(17) com um tempo limite de 1 hora, respectivamente. A coluna *Gap-MM(%)* reporta o gap entre o valor da solução da heurística e o valor da solução do modelo matemático. Os valores de *Gap-LB(%)* e *Gap-MM(%)* foram calculados conforme as Equações (18) e (19).

$$Gap\text{-}LB(\%) = 100(Custo - LB)/Custo \tag{18}$$

$$Gap\text{-}MM(\%) = 100(CustoHeur - CustoMM)/CustoMM \tag{19}$$

Da Tabela 3, observa-se que o modelo matemático, dentro do tempo limite considerado (1h), conseguiu provar a otimalidade de apenas uma solução, enaltecendo a dificuldade de resolução

DOI: 10.59254/sbpo-2023-175184
http://live-proceedings.pantheonsite.io/p/175184

Tabela 3: Resultados preliminares.

Inst.	$\|T\|$	Modelo Matemático				Heurística R&F e F&O			
		Custo	CPU T(s)	LB	Gap-LB(%)	Custo	CPU T(s)	Gap-LB(%)	Gap-MM(%)
#1	4	309,457.61	2441.09	309,457.61	0.00	309,457.61	298.20	0.00	0.00
#2	5	1,287,280.63	3600.00	1,182,679.82	8.13	1,287,841.96	236.89	8.17	0.04
#3	6	1,403,849.32	3600.00	1,262,574.37	10.06	1,405,454.12	429.24	10.17	0.11
#4	11	4,439,812.38	3600.00	3,792,488.38	14.58	4,439,236.72	1391.33	14.57	-0.01
#5	21	8,864,262.44	3600.00	6,675,232.38	24.70	8,862,739.80	3600.00	24.68	-0.02

do problema. As soluções obtidas com as heurísticas são bem próximas daquelas obtidas com o modelo matemático. Nas instâncias com $|T| \in \{4, 5, 6\}$, as soluções são ligeiramente piores do que as retornadas pelo modelo matemático, porém os tempos de processamento são significativamente menores (*CPU T(s)* < 430s). Nas instâncias maiores, com $|T| \in \{11, 21\}$, as heurísticas retornaram soluções melhores. A solução para a instância *#4* foi apenas 0.01% melhor, porém com menos da metade do tempo utilizado pelo modelo matemático, enquanto para a instância *#5* a solução obtida foi 0.02% melhor com o mesmo tempo de processamento.

Com base nos resultados apresentados, uma pergunta natural é: *se fossem adotados outros tempos limites, qual abordagem retornaria o melhor resultado?* Para responder essa pergunta, foram realizados gráficos de convergência considerando um tempo limite de 1 hora, conforme ilustrados na Figura 4. A partir dos gráficos de convergência, percebe-se que o modelo (1)-(17) tem uma convergência mais rápida para as melhores soluções, quando comparado com a resolução com a heurística R&F-F&O. A heurística, apesar de ter uma convergência mais lenta, melhora gradativamente durante a sua execução, terminando por encontrar soluções equivalentes às obtidas pelo modelo matemático. Portanto, para tempos limites mais restritos, o uso do modelo (1)-(17) seria o mais indicado, enquanto para tempos limites maiores, o uso da heurística R&F-F&O seria o mais indicado, especialmente para instâncias maiores.

6. Conclusões

Neste artigo foram apresentados um modelo matemático e uma heurística baseada em horizonte deslizante para resolução de um problema de transferência e transportes de valores inspirados em um caso real enfrentado por um banco espanhol com atuação no Peru. Para avaliar o desempenho na resolução do modelo matemático e da heurística propostos, cinco instâncias com horizonte de tempo variando de 4 a 21 períodos foram resolvidas. Verificou-se que a utilização do modelo matemático é o mais indicado para resolver instâncias menores com tempos limites mais restritos, enquanto a heurística seria mais indicada para lidar com instâncias com horizontes de tempo mais longos e tempos limites maiores (e.g., $>$ 1h).

Como sugestão de trabalhos futuros, podemos listar: i) a utilização de algoritmos de corte para melhorar a eficiência da resolução dos modelos matemáticos, ii) a avaliação de outros valores de parâmetros para melhorar a eficiência da heurística R&F-F&O, e iii) a consideração de demandas estocásticas para uma maior aproximação com a realidade.

Agradecimentos

O presente trabalho foi realizado com apoio parcial da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

DOI: 10.59254/sbpo-2023-175184
http://live-proceedings.pantheonsite.io/p/175184

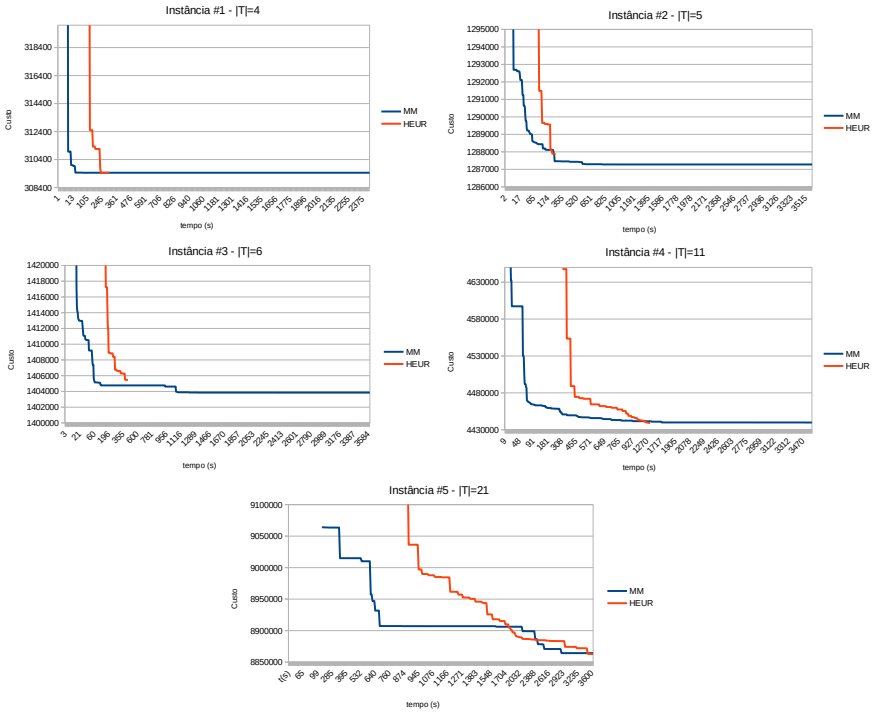


Figura 4: Gráficos de convergência

Referências

Apolinário, L. A. (2020). Otimização de limitantes superiores para o problema de dimensionamento de lotes em máquinas paralelas. Master's thesis, Universidade Federal da Paraíba, João Pessoa.

Batı, Ç. e Gözüpek, D. (2019). Joint optimization of cash management and routing for new-generation automated teller machine networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(12):2724–2738.

Diao, Y., Sarkar, R., e Jan, E.-E. (2016). Optimizing atm cash flow network management. In *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, p. 1073–1078.

Özer, F. (2019). Comparison of Integer Linear Programming and Dynamic Programming Approaches for ATM Cash Replenishment Optimization Problem. Master's thesis, Computer Engineering Department, Middle EastTechnical University.

van Anholt, R. (2014). *Optimizing Logistics Processes in Cash Supply Chains*. PhD thesis, Vrije Universiteit Amsterdam.

Zhou, Y., Wang, Q., Wu, Y., e Huang, M. (2020). Data-Driven Cash Replenishment Planning of Recycling ATMs with Out-of-Cash and Full-of-Cash Risks. *International Journal of Information Systems and Supply Chain Management*, 13(2):77–96.

DOI: 10.59254/sbpo-2023-175184
http://live-proceedings.pantheonsite.io/p/175184