LII Simpóso Brasileiro de Pesquisa Operacional
João Pessoa-PB, 3 a 5 de novembro de 2020

UMA PROPOSTA BASEADA EM UM MODELO DE PLI PARA MITIGAR A RUPTURA DO VAREJO FARMACÊUTICO

Frederico Lopes Fiuza
Instituto Federal Fluminense - IFF
Rua Dr. Siqueira, 273. Parque Dom Bosco - Campos dos Goytacazes - RJ. CEP: 28030-130
frederico.fiuza@gmail.com

Frederico Galaxe Paes
Instituto Federal Fluminense - IFF
Av. Souza Mota, 350. Parque Fundão - Campos dos Goytacazes - RJ. CEP: 28060-010
fpaes@iff.edu.br

Carlos Leonardo Ramos Póvoa
Universidade Estadual do Norte Fluminense - UENF
Av. Alberto Lamego, 2000. Parque Califórnia - Campos dos Goytacazes - RJ. CEP: 28013-602
clrp@uenf.br

André Soares Velasco
Instituto Federal Fluminense - IFF
Av. Souza Mota, 350. Parque Fundão - Campos dos Goytacazes - RJ. CEP: 28060-010
asvelasco@iff.edu.br

RESUMO

A ruptura do varejo, isto é, quando o cliente não encontra o produto desejado, é um dos maiores causadores de evasão de clientes. O mercado farmacêutico em plena expansão não está ileso a esse problema que é comum as cadeias de estabelecimentos varejistas. O objetivo deste trabalho é propor uma solução para mitigar a ruptura do varejo farmacêutico com o propósito de garantir uma compra mais assertiva para o cliente. Foi elaborado um modelo de Programação Linear Inteira (PLI) para obter a alocação ótima dos produtos às farmácias, minimizando o custo da cesta de produtos. O modelo foi resolvido utilizando o *solver* IBM ILOG® CPLEX®, sendo encontradas as soluções ótimas para todas as instâncias testadas com tempos de processamento inferiores a um segundo, validando o modelo proposto.

Palavras-chave: **Mercado Farmacêutico; Varejo de Farmácias; Programação Linear Inteira**.

ABSTRACT

The breakdown of the retail, when the customer does not find the desired product, is one of the biggest cause of customer evasion. The expanding pharmaceutical market is not unscathed from this problem common to other chains of retail establishments. The objective of this work was to propose a solution to mitigate the rupture of pharmaceutical retail in order to guarantee a more assertive purchase for the client. A integer linear programming model was developed to obtain the optimal allocation of products to pharmacies, minimizing the cost of the products. The model was solved using IBM® ILOG® CPLEX® solver, and the optimal solution has been found for all tested instances with prossessing times less than one second, validating the proposed model.

Key words: **Pharmaceutical Market; Pharmaceutical retails; Integer Linear Programming.**

Galoá { Este trabalho foi publicado utilizando o **Galoá** proceedings

1. Introdução

O crescimento do mercado farmacêutico nas últimas décadas tem sido impulsionado pelos medicamentos genéricos [Gomes 2014]. Além disso, o Brasil possui um elevado gasto com medicamentos, principalmente nas famílias de menor renda [Fiuza e Lisboa 2001], causando um impacto significativo no orçamento de milhões de lares e criando um cenário propício para a livre concorrência de mercado, onde a busca pelo medicamento que esteja disponível pelo menor preço, se torna cada vez mais frequente e relevante.

Diante disso, o diferencial de mercado está nos estabelecimentos que possuam o item desejado a pronta entrega, pelo melhor preço e que seja facilmente adquirido através de seus canais de vendas.

O mercado farmacêutico, apesar de ser maduro, não está livre de problemas comuns da cadeia do varejo, tais como: atraso de entrega, falta de matéria prima, rupturas de estoque (*stock-out*), sendo que esse último é um dos que causa maior evasão de clientes à concorrência.

Estima-se que quase 60% dos clientes vão procurar em concorrentes quando existe ruptura de estoque e que, aproximadamente, apenas 3% desistam da compra [Gouvêa e Biazzi 2007]. Assim, quando um cliente desse mercado toma a decisão de consumir um produto, ele irá fazer em mais de 90% dos casos.

A indústria farmacêutica mundial vem passando por mudanças importantes no que se refere ao perfil de demanda dos produtos, nas competências tecnológicas e no cenário competitivo, motivando as empresas do setor a adotar novas estratégias [Gomes et al. 2014].

O Brasil está inserido no contexto global da indústria farmacêutica como um dos mercados mais dinâmicos da última década, onde apresenta taxas de crescimento de dois dígitos. Tal crescimento foi impulsionado pelos medicamentos genéricos e pela ascensão de novos consumidores [Gomes et al. 2014].

No estudo de [Gouvêa e Biazzi 2007], onde 106 consumidores de uma drogaria foram entrevistados, obtiveram o seguinte resultado ao serem questionados sobre seu comportamento na situação de não haver encontrado o produto desejado na drogaria: 59,3% procurariam em outra drogaria; 21,3 % comprariam produto genérico; 2,8 % apenas desistiriam da compra.

Desenvolver uma solução que permita compor uma cesta de produtos farmacêuticos otimizada é fundamental para o consumidor, uma vez que esta lhe garantirá uma compra com menor custo e com maior comodidade. Além disso, as farmácias terão acesso a uma solução que permita uma maior visibilidade de venda e competitividade de mercado.

Portanto, este trabalho tem como objetivo propor uma solução baseada em um aplicativo para *smartphone* que garanta ao cliente uma maior confiabilidade de encontrar os produtos que deseja proveniente de estabelecimentos farmacêuticos, através de um canal de venda que proporciona mais comodidade e pelo menor preço. Para otimizar a compra, foi formulado um modelo de Programação Linear Inteira (PLI) que obtém a alocação ótima dos produtos às farmácias, minimizando o custo da cesta de produtos.

Assim sendo, o modelo proposto pretende responder as seguintes perguntas: como garantir que o cliente sempre achará o(s) produto(s) desejado(s) pelo menor custo? Ou ainda, como alocar a compra do(s) produto(s) desejado(s) à(s) farmácia(s) de modo que se tenha esse menor custo, respeitando as restrições de estoque e de entrega das mesmas?

O presente artigo está organizado da seguinte maneira: Na Seção 2 é feito uma revisão da literatura, de modo a embasar o estudo. Na Seção 3 são apresentados os procedimentos metodológicos, bem como uma breve descrição da arquitetura do sistema proposto. O modelo matemático proposto é descrito na Seção 4. Na Seção 5 são demonstrados os resultados dos testes computacionais conduzidos com as instâncias geradas e, finalmente, a Seção 6 apresenta as considerações finais.

2. Revisão da literatura

Dentre os diversos trabalhos encontrados envolvendo otimização, destacam-se os relacionados ao problema de alocação, mas que não abordam diretamente o problema de alocação

Galoá { Este trabalho foi publicado utilizando o **Galoá** proceedings

de produtos a farmácias/drogarias. Alguns destes trabalhos, que estão de alguma forma relacionados ao objeto de estudo, são descritos a seguir.

Em [Assis 2014] é estudado o problema de alocação de chaves (PAC) em sistemas radiais de distribuição de energia elétrica. Uma metodologia baseada em um Algoritmo Memético, juntamente com uma população estruturada é proposta para alocar chaves seccionadoras e de manobra, manuais e automáticas, com diferentes capacidades.

[Freiras Filho 2016] apresenta uma proposta de solução para o Problema de Alocação de Professores em Disciplinas, levando em consideração o processo adotado pela Universidade Federal do Ceará e suas restrições. Para resolver o problema, foi formulado e implementado um modelo de PLI e realizados experimentos com dados aleatórios e com dados reais para análise do modelo.

Em [Yu et al. 2017], o tradicional Problema de Alocação de Portões em Aeroportos (*Airport Gate Assignment Problem* - AGAP) é estendido, onde os custos tradicionais e a robustez são simultaneamente considerados. Para resolver o complexo modelo não linear gerado, os autores propõem um algoritmo de busca local em vizinhança de grande porte adaptativo (*Adaptive Large Neighborhood Search* – ALNS). Comparações com algoritmos de referência da literatura mostram a competitividade do algoritmo proposto para resolver o AGAP.

Em [Sanchotene 2018] é tratado o Problema de Alocação de Médicos do Hospital de Clínicas de Porto Alegre, onde é implementado um método heurístico para solução do problema do hospital. A heurística, então, é comparada com métodos exatos e com um método baseado em programação matemática.

[Faudzi et al. 2020] resolvem o problema de alocação de alunos estagiários a professores supervisores, maximizando as preferências totais declaradas pelos estudantes. Um modelo de PLI é desenvolvido considerando restrições que envolvem a capacidade do professor em função de sua posição acadêmica e associação de gênero aluno-professor. O estudo demonstrou que o método proposto é eficiente, evitando ajustes na solução.

[Sindi e Agbelie 2020] lidam com o problema de atribuir serviços de manutenção específicos em um sistema de gestão de pavimentação de uma rede de estradas. Os autores resolvem o problema através de uma metodologia de otimização multiobjetivo que utiliza algoritmos genéticos (AG) e programação linear inteira mista (PLIM). Os resultados mostram que o método proposto é capaz de alocar serviços de manutenção às vias eficientemente, maximizando a vida útil do serviço a um custo mínimo.

3. Procedimentos metodológicos

O problema de encontrar a alocação ótima de um conjunto de produtos com demandas diferentes a um conjunto de fornecedores (farmácias) de modo a minimizar os custos envolvidos, surge quando se desenvolve um sistema de *delivery* por aplicativo em *smartphones*. Assim, a metodologia de pesquisa adotada inclui a revisão bibliográfica sobre diversos problemas de otimização envolvendo alocação/atribuição, uma vez que estes problemas estão estritamente relacionados ao problema proposto. Com o escopo definido na revisão, foi proposto uma solução que tem como ponto central um aplicativo de *smartphone* que garanta ao cliente a confiabilidade de encontrar os produtos desejados com menor preço.

3.1 Aplicativo Pediufarma®

Mediante as pesquisas realizadas, bem como as análises e estudos de mercado mostrando a viabilidade de exploração deste nicho, foi proposto o desenvolvimento do aplicativo para *smartphones* Pediufarma®. Trata-se de um ecossistema de soluções tecnológicas que implementa parte do que é proposto neste artigo.

Sua operação efetiva teve início em março de 2020 e com dois meses de atividades já havia alcançado mais de sete mil *downloads*, tendo como ponto inicial de operação a cidade de Campos dos Goytacazes – RJ. Atualmente, são mais de 50 unidades de farmácias de diversos portes operando através do ecossistema, indo desde farmácias pequenas de bairros a redes de franquias

nacionais. A cidade de Macaé – RJ iniciou o processo de inclusão no aplicativo, estando em fase inicial e operando desde o mês de Abril de 2020. Até o final do ano há uma previsão do sistema estar operando em no mínimo 15 cidades de diversos estados.

3.2 Arquitetura tecnológica

A divisão do problema foi feita em quatro partes menores, conforme descrito na Figura 1.



Figura 1 – Etapas para solução.

1. **Coletar a demanda:** etapa responsável por obter do cliente os itens os quais deseja comprar, bem como a localização onde o cliente deseja que seja entregue o seu pedido;
2. **Preparar a cesta de produtos:** etapa responsável por buscar todos os estabelecimentos elegíveis, bem como as restrições inerentes a cada negócio (raio de entrega, valor de frete, valor mínimo de pedido e disponibilidade do produto), a fim de atender a demanda do cliente. Cabe ressaltar que o critério responsável por eleger um estabelecimento é possuir o item em estoque e o cliente estar dentro da área de abrangência do estabelecimento;
3. **Otimizar a cesta:** etapa responsável em otimizar a cesta de produtos dentre todos os estabelecimento elegíveis que foram selecionados, respeitando as restrições de valor do frete e pedido mínimo aceitável;
4. **Distribuir demanda:** etapa responsável por notificar os estabelecimentos sobre a demanda do cliente.

Uma vez definido o problema maior e dividindo-o em partes menores, resta definir como tais partes serão aninhadas com o intuito de facilitar o seu desenvolvimento e manutenção. Seguindo a premissa *"Single Responsability Principle"*, que em programação defende que uma classe deve possuir apenas uma responsabilidade ou ação para executar [Martin 2000], cada etapa foi dividia para seus respectivos servidores, sendo as informações armazenadas em um banco de dados. A este cenário agregam-se o aplicativo do cliente (Pediufarma®) que é onde o processo de pedido se inicia e um sistema instalado nos computadores das farmácias parceiras, responsável por notificar a demanda e onde o processo finaliza, conforme descrito na Figura 2. Dessa forma, são listadas as seguintes ferramentas tecnológicas desenvolvidas para a solução:

1. Aplicativo do cliente;
2. Servidor *web* coletor de demandas;
3. Servidor principal;
4. Servidor de banco de dados;
5. Servidor para otimização de preços;
6. Servidor para distribuir demandas;
7. Sistema notificador dos estabelecimentos;

O ponto de partido do problema é o aplicativo, no qual o cliente, mediante prévio cadastro, consegue informar sua localização e montar uma lista de produtos e farmácias que contém a sua demanda. Uma vez coletado, o cliente envia a sua demanda para o coletor de demandas que aplica regras de segurança e validação, que dentre outras funções, garante a integridade dos dados e a segurança do mesmo. O coletor de demandas aciona um servidor principal que é capaz de acessar o banco de dados e extrair dali apenas os estabelecimentos elegíveis para atender a demanda

do cliente. A elegibilidade de um estabelecimento se dá mediante ao cumprimento obrigatório e indispensável de dois critérios:

1. **Possuir estoque**: para mitigar a ruptura do varejo só são elegíveis estabelecimentos que possuem o item em estoque;
2. **Entrega disponível**: o estabelecimento garante a entrega na localização do cliente.

Com os estabelecimentos elegíveis filtrados, o servidor principal aciona o servidor que será responsável por otimizar a cesta. Esse servidor calcula os dados de entrada da instância levando em consideração os preços dos produtos nas farmácias, a quantidade demandada de cada produto, os fretes cobrados pelos estabelecimentos, as quantidades disponíveis nas farmácias dos produtos solicitados e os valores dos pedidos mínimos para entrega praticados pelas farmácias.

Após a instância ser construída, um problema de PLI é montado e resolvido por um *solver*. Existem diversos resolvedores disponíveis, alguns proprietários necessitando de licença para uso comercial (e.g. CPLEX, Gurobi e Xpress), outros gratuitos (e.g. COIN-OR e GLPK). De posse da solução exata do problema, o servidor otimizador encaminha para o servidor principal montar a cesta de produtos, notificar e informar ao cliente a cesta e o seu respectivo valor. O cliente aceitando a proposta cabe ao servidor principal acionar o distribuidor de demandas.

O distribuidor de demandas envia a notificação para cada estabelecimento mediante um *software* instalado nos mesmos aguardando serem demandadas. Cada farmácia receberá apenas a sua demanda bem como os dados do cliente, sua localização e a forma que o cliente deseja pagar. Ressalta-se que numa mesma cesta pode haver *n* produtos de *n* farmácias, porém, as unidades de um determinado produto são direcionadas para uma única farmácia, conforme restrição (2) do modelo da Seção 4. Uma vez distribuído a demanda, o cliente é informado no aplicativo que seu pedido foi enviado, bem como cada farmácia, concluindo-se assim a solução do problema.

Para que esse processo funcione, fica sob a responsabilidade das farmácias manterem a sua tabela de preços atualizada no servidor de banco de dados, processo que é feito de forma manual ou de forma automática combinado com os estabelecimentos.

Na Seção 4 será descrita a formulação do modelo de PLI a ser implementado no servidor. Para testar o modelo, considerando o tamanho das instâncias em função da média de pedidos, diversos testes computacionais foram conduzidos com instâncias aleatórias geradas a partir de dados reais (Seção 5).

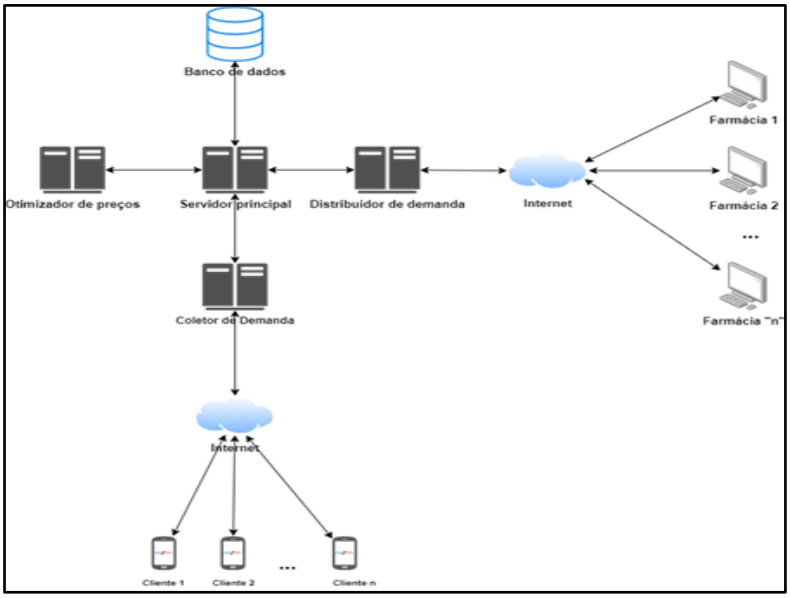


Figura 2 – Arquitetura do sistema.

É importante destacar que além do preço dos produtos, diversos outros parâmetros como o valor mínimo do pedido, disponibilidade do produto e frete também compõem o problema aumentando ainda mais sua complexidade.

4. Modelo matemático proposto

Dados os conjuntos $P = \{p_1, p_2, ..., p_i, ..., p_n\}$ de n produtos e $D = \{f_1, f_2, ..., f_j, ..., f_m\}$ de m farmácias, o objetivo é encontrar a alocação ótima dos n produtos às m farmácias, tal que a demanda d_i de cada produto i seja atribuída a uma única farmácia j respeitando a capacidade de estoque Q_{ij} que a farmácia j tem do produto i. Além disso, o valor dos produtos atribuídos à farmácia j não pode ser inferior ao valor do pedido mínimo PM_j para *delivery* estipulado pela mesma. A alocação ótima é dada pela minimização da soma dos preços dos produtos atribuídos às farmácias, mais o valor do frete F_j que é cobrado por algumas farmácias (Figura 3).

Cabe ressaltar que os produtos de um mesmo tipo só podem ser atribuídos a uma única farmácia, porém, produtos diferentes podem ser atribuídos a farmácias diferentes.

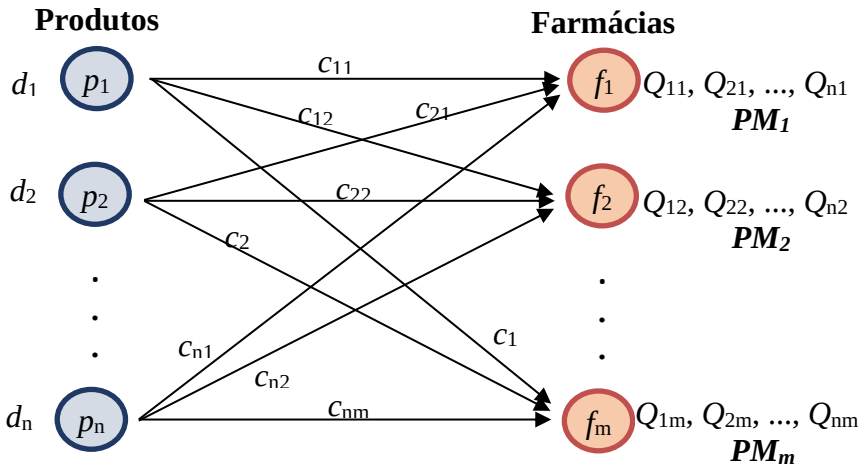


Figura 3 – Parâmetros envolvidos em uma alocação de n produtos a m farmácias.

O modelo proposto para resolver o problema de alocação produtos-farmácias é uma formulação de Programação Linear Inteira 0 – 1, cujo objetivo é minimizar o custo de aquisição da cesta de produtos, respeitando as restrições do problema. Para a formulação do modelo matemática proposto para o problema, algumas notações devem ser consideradas.

Conjuntos:

- $I = \{1, 2, ..., i, ..., n\}$: conjunto de produtos, onde i é o índice de cada tipo de produto;
- $J = \{1, 2, ..., j, ..., m\}$: conjunto de farmácias, onde j é o índice de cada farmácia;

Dados:

- c_{ij}: custo do produto i na farmácia j;
- d_i: demanda do produto i;
- F_j: frete cobrado pela farmácia j. Caso a farmácia não cobre frete, $F_j = 0$;
- Q_{ij}: quantidade do produto i em estoque na farmácia j;

- PM_j: valor do pedido mínimo para entrega praticado pela farmácia j.

Variáveis:

- $x_{ij} = \begin{cases} 1\,, & \text{se o produto } i \text{ é comprado na farmácia } j; \\ 0\,, & \text{caso contrário;} \end{cases}$
- $y_j = \begin{cases} 1\,, & \text{se a farmácia } j \text{ é selecionada;} \\ 0\,, & \text{caso contrário;} \end{cases}$

$$Z = \text{Min}\left(\sum_{j\in J}\sum_{i\in I} x_{ij}c_{ij}\,d_i + \sum_{j\in J} y_j F_j\right) \qquad (1)$$

$s.a.:$

$$\sum_{j\in J} x_{ij} = 1 \qquad \forall i \in I \qquad (2)$$

$$x_{ij}d_i \leq Q_{ij} \qquad \forall i \in I, \forall j \in J \qquad (3)$$

$$\sum_{i\in I} x_{ij}c_{ij}d_i \geq y_j PM_j \qquad \forall j \in J \qquad (4)$$

$$x_{ij} \leq y_j \qquad \forall i \in I, \forall j \in J \qquad (5)$$

$$x_{ij} \in \{0,1\},\ y_j \in \{0,1\} \qquad \forall i \in I, \forall j \in J \qquad (6)$$

Neste modelo, a função objetivo (1) minimiza a soma dos custos de atribuição dos produtos às farmácias, mais os custos do frete eventualmente cobrados pelas mesmas. As restrições (2) garantem que cada tipo de produto seja atribuído ou comprado em uma única farmácia. As restrições (3) garantem que a farmácia *j* tenha estoque suficiente para atender a demanda do produto *i*, caso ela seja selecionada. As restrições (4) garantem que a farmácia *j* só será selecionada se o preço dos produtos atribuídos a ela ultrapassar o preço mínimo de venda PM_j. As restrições (5) garantem que só haverá alocação a uma determinada farmácia, se esta for selecionada. Finalmente, as restrições (6) definem as variáveis do modelo x_{ij} e y_j como binárias.

5 Testes computacionais

Para a validação do modelo matemático proposto, testes computacionais foram realizados em 10 instâncias geradas aleatoriamente usando dados fictícios, instâncias F1_n1m3 a F10_n30m40 (Tabela 2), e duas instâncias usando dados reais, instâncias R1_n10m20 e R2_n15m20 (Tabela 2). O modelo foi implementado utilizado o *software* de otimização IBM® ILOG® CPLEX® *Optimization Studio* v12.9 (CPLEX), que possui uma interface de programação (API) orientada a objetos para a linguagem C++ conhecida como *Concert Technology*. Os testes computacionais foram conduzidos em uma máquina Intel Core $i7-3517U$, 1.9 GHz CPU, 6 GB RAM, rodando em um sistema operacional Windows 10.

A Tabela 2 apresenta as quantidades de produtos *n* e de farmácias *m* das doze instâncias geradas, enquanto a Figura 4 apresenta os dados da Instância F5_n7m10 (n = 7, m = 10).

Galoá { Este trabalho foi publicado utilizando o **Galoá** proceedings

Tabela 2 – Tamanhos das instâncias geradas.

Instância	nº de Produtos (n)	nº de Farmácias (m)
F1_n1m3	1	3
F2_n1m10	1	10
F3_n3m3	3	3
F4_n3m10	3	10
F5_n7m10	7	10
F6_n7m20	7	20
F7_n10m20	10	20
F8_n10m40	10	40
F9_n30m20	30	20
F10_n30m40	30	40
R1_n10m20	10	20
R2_n15m20	15	20

n(tipos diferentes de produtos): 7
m(quantidade de farmácias): 10

Custo do produto i(linha) na farmácia j(coluna) (c_ij):

i\j	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	2.3	2.5	2	3.7	3	1.8	1.8	2.3	2.3	2.5
2	10	11	10.1	9.9	9	8	11	10.4	11.2	9.8
3	15	14.5	15.6	14	13	15.8	16	16.8	16.2	15.9
4	22	23	22.6	22.1	22.8	23.1	23.6	23.9	24	25
5	55	44	49	49	47	45	48	45	55	49
6	34.5	37.9	38.6	37.2	37.3	42.8	41.8	35.7	33.9	37.7
7	60.6	77.3	77.4	67.2	63.3	76.8	77.7	69.8	77.7	69.7
F_j	10	15	10	0	15	10	0	0	15	20
PM_j	10	2	10	0	15	10	0	10	8	2

Estoque do produto i(linha) na farmácia j(coluna) (Q_ij):

i\j	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**d_i**
1	11	13	0	11	10	0	0	11	11	12	1
2	10	11	15	10	12	0	14	13	11	0	2
3	8	9	0	10	10	0	0	0	8	0	1
4	7	6	7	0	5	0	5	8	6	8	3
5	0	7	8	0	0	0	7	5	7	6	2
6	2	2	2	3	1	1	0	0	1	2	1
7	1	2	1	1	0	0	2	3	3	1	1

Figura 4 – Dados da Instância F5_n7m10.

É importante destacar que uma instância ao ser gerada só deve considerar os produtos disponíveis em estoque em pelo menos uma farmácia. Caso um ou mais produtos estejam em falta em todas as farmácias, este(s) deve(m) ser eliminado(s) da lista de produtos. Isso evita que a solução da instância seja inviável.

De modo a assegurar somente soluções viáveis, foi implementado uma etapa de pré-processamento, cujo objetivo é evitar a situação descrita no parágrafo anterior. Deste modo, antes do modelo ser gerado e resolvido é verificado se para cada produto solicitado ao menos uma farmácia atenda a demanda do respectivo produto. Se todos os produtos puderem ser entregues, o

modelo é resolvido normalmente, caso contrário, uma mensagem indicando a indisponibilidade do(s) produto(s) é disparada antes de prosseguir para a solução do modelo.

Tabela 3 – Resultado dos testes computacionais.

Instância	Z	t(s)
F1_n1m3	33	0,024
F2_n1m10	14,4	0,038
F3_n3m3	47,2	0,049
F4_n3m10	37,2	0,040
F5_n7m10	297,2	0,049
F6_n7m20	289,2	0,050
F7_n10m20	825,5	0,049
F8_n10m40	825,5	0,068
F9_n30m20	3869,66	0,075
F10_n30m40	3842,52	0,091
R1_n10m20	237,86	0,086
R2_n15m20	591,36	0,082

De acordo com a Tabela 3, todas as instâncias levaram menos que um segundo para serem resolvidas. Mesmo as Instâncias F9_n30m20 e F10_n30m40, criadas para testar a robustez do modelo matemático, embora ocorram com menos frequência na prática, retornaram a solução ótima em menos de um segundo.

Um segundo tipo de teste realizado foi para simular o desempenho do servidor otimizador, cujo objetivo foi tentar medir o maior tempo de espera o qual um cliente pode ser submetido ao fazer um pedido, considerando um grande número de requisições/chamadas simultâneas. Este teste é importante, pois simula o comportamento do aplicativo Pediufarma® em uma situação de acesso concorrente crescente, algo provável de acontecer na medida em que o número de usuários aumente. Neste teste, mediu-se apenas o tempo que o CPLEX levou para resolver um determinado número de requisições simultâneas, considerando instâncias com tamanhos aleatórios.

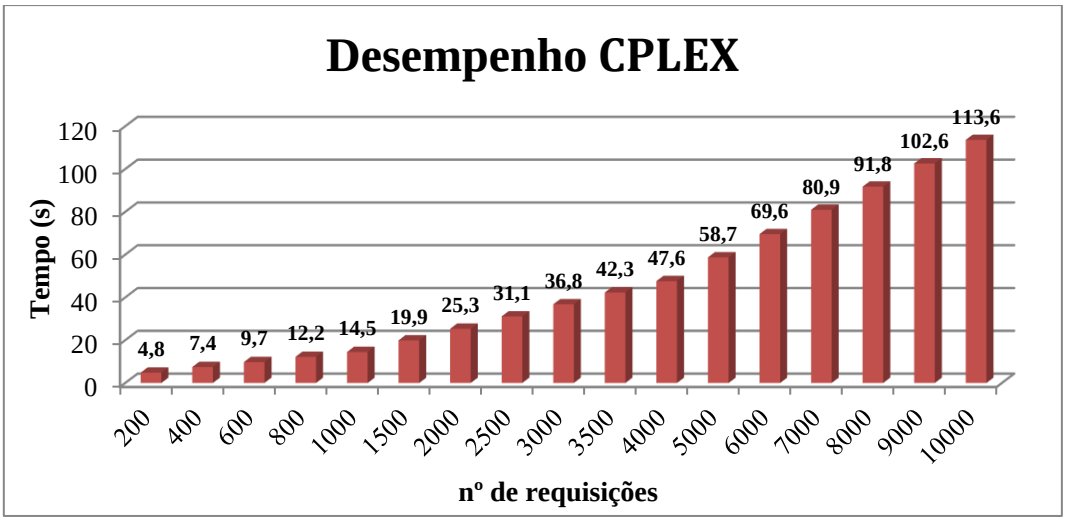


Figura 5 – Tempo de resposta em função do número de chamadas do CPLEX.

O teste foi realizado em uma máquina Intel Core *i*7 − 8565U, 1,99GHz CPU, 4-CORE, 8 processadores lógicos, 16GB RAM, simulando um servidor dedicado. A Figura 5 apresenta os tempos em segundos para diferentes quantidades de requisição. Portanto, para uma quantidade de 5000 requisições sequenciais, o último pedido seria atendido após um tempo de aproximadamente 58,7 segundos (menos que 1 minuto).

Destaca-se que um volume de 1000 requisições por segundo seria um volume alto de tráfego do aplicativo, sendo que o tempo de resposta do último cliente seria de menos de 15 segundos (Figura 5). Todos os tempos mensurados não consideram o tempo de transmissão do servidor para o aplicativo. Se o volume de clientes aumentar, a arquitetura desenvolvida permitiria incorporar mais de um servidor para resolver o modelo, fazendo que o tempo de espera do usuário diminua consideravelmente.

6. Considerações finais

A pesquisa propõe um processo que possa melhorar a interação do cliente com os estabelecimentos farmacêuticos, de forma que a ruptura do varejo seja mitigada e a compra possa ocorrer no menor tempo, com menor preço e na maior comodidade, valendo-se de um modelo de otimização de preços que afiance uma cesta de compra mais barata.

O modelo matemático proposto foi capaz de obter a solução ótima em todas as instâncias testadas com tempo inferior a um segundo. Dessa forma, pode-se dizer que o modelo está habilitado a rodar em um servidor dedicado e, assim, ser incorporado ao aplicativo.

A próxima etapa da pesquisa consiste em incorporar o módulo otimizador ao sistema proposto, bem como a realização de testes computacionais adicionais para fornecer o tempo médio de resposta do servidor para o usuário do aplicativo. Esse tempo será importantes para dimensionar a capacidade de processamento dos servidores responsáveis em resolver o modelo e, como consequência, o tempo de espera do cliente ao fazer um pedido pelo aplicativo Pediufarma®.

Referências

ABPMP. BPM CBOK *Guia para o Gerenciamento de Processos de Negócio*. 1a ed. Brasil: Association Of Business Process Management Professionals, 2013.

ASSIS, L. S. D. (2014). *Otimização de alocação de chaves em redes de distribuição de energia elétrica*. Tese (Doutorado em Engenharia Elétrica) – Campinas – SP, Universidade Estadual de Campinas – Unicamp.

BRITO, M. O que é Marketplace? - Vantagens e Desvantagens | MZclick, 2018. Disponível em: <https://www.mzclick.com.br/o-que-e-marketplace-vantagens-e-desvantagens/>. Acesso em: 19 dez. 2018

COUGHLAN, A. T. et al. Canais de Marketing. Edição: 1 ed. [s.l.] Pearson Universidades, 2011.

FAUDZI, S., ABDUL-RAHMAN, S., ABD RAHMAN, R., ZULKEPLI, J., & BARGIELA, A. (2020). Optimizing the Preference of Student-Lecturer Allocation Problem Using Analytical Hierarchy Process and Integer Programming. *Journal of Engineering Science and Technology*, *15*(1), 261-275.

FIÚZA, E. P. S.; LISBOA, M. B. (2001). *Bens credenciais e poder de mercado: Um estudo econométrico da indústria farmacêutica Brasileira*. Technical report, Texto para Discussão do IPEA, Rio de Janeiro.

Galoá { Este trabalho foi publicado utilizando o **Galoá** proceedings

FREITAS FILHO, F. S. D. (2016). *Uma abordagem para o problema de alocação de professores em disciplinas utilizando programação linear inteira*. Trabalho de Conclusão de Curso – Quixadá - CE, Universidade Federal do Ceará – UFCE.

GOMES, R. et al. O novo cenário de concorrência na indústria farmacêutica brasileira. abr. 2014.

GOUVÊA, M. A.; BIAZZI, M. R. DE. O comportamento do consumidor do setor farmacêutico diante da demanda não atendida – implicações para a cadeia de suprimentos. *Revista de Administração Mackenzie* (Mackenzie Management Review), v. 8, n. 1, 2007.

MARTIN, R. C. Design Principles and Design Patterns. p. 34, 2000.

SANCHOTENE, T. C. (2018). *Abordagem heurística para solução do problema de alocação de médicos do HCPA*. Trabalho de Conclusão de Curso – Porto Alegre – RS, Universidade Federal do Rio Grande do Sul – UFRGS.

SINDI, W., AGBELIE, B. (2020). Assignments of Pavement Treatment Options: Genetic Algorithms versus Mixed-Integer Programming. *Journal of Transportation Engineering, Part B: Pavements*, *146*(2), p. 04020008.

TRADEFORCE. 4 ações para ter maior controle de ruptura no varejo, 19 jun. 2018. Disponível em: <https://accera.com.br/blog/4-acoes-para-ter-maior-controle-de-ruptura-no-varejo/>. Acesso em: 29 mar. 2020

Yu, C., Zhang, D., Lau, H. Y. (2017). An adaptive large neighborhood search heuristic for solving a robust gate assignment problem. *Expert Systems with Applications*, *84*, 143-154.

Galoá { Este trabalho foi publicado utilizando o **Galoá** proceedings