



ALGORITMOS GENÉTICOS APLICADOS AO PROBLEMA DE LOCALIZAÇÃO DE ARMAZÉNS CAPACITADOS POR MEIO DE UMA ABORDAGEM DE OTIMIZAÇÃO E VISUALIZAÇÃO SOB DEMANDA

Flavio Oliveira Milazzo¹, Alexandre Cardoso¹, Claudianne Myllene Morais de Almeida¹, Edgard Lamounier¹, Lara Cristina Resende Silva Couto¹, Keiji Yamanaka¹, Lucas Severino Dos Santos Teixeira¹
¹FEELT – Universidade Federal de Uberlândia

Resumo - Definir a estratégia de distribuição de uma cadeia de abastecimento é importante para obter um planejamento eficiente dos locais de armazenagem e das rotas de transportes de mercadorias. Assim, as empresas tradicionalmente investem em construção e/ou aluguel de novos armazéns ou contratação de operadores logísticos com o intuito de se aproximarem dos mercados consumidores. Em função do aumento do comércio eletrônico, novos modelos de negócio combinam espaços subutilizados de empresas com clientes que necessitam de espaço extra de armazenagem, ampliando a flexibilidade e proporcionando alternativas ao modelo tradicional de construção ou aluguel de armazéns. Dessa forma, o presente artigo propõe uma abordagem, sob demanda, de otimização e visualização da solução para o problema de localização de armazéns capacitados, utilizando algoritmos genéticos e programação linear com o objetivo de encontrar a solução ótima ou se aproximar da mesma, para um estudo de caso aplicado ao comércio eletrônico brasileiro.

Palavras-Chave- armazenagem sob demanda, problema de localização de armazéns capacitados, algoritmos genéticos, DEAP, programação linear, MILP

GENETIC ALGORITHMS APPLIED TO THE CAPACITATED WAREHOUSE LOCATION PROBLEM THROUGH AN APPROACH OF OPTIMIZATION AND VISUALIZATION ON DEMAND

Abstract - Defining the distribution strategy of a supply chain is important for efficient planning of storage locations and freight transport routes. Thus, companies traditionally invest in building and/or renting new warehouses or hiring logistics operators in order to get closer to consumer markets. Due to the increase in e-commerce, new business models combine underused spaces of companies with customers who need extra storage space, increasing flexibility and providing alternatives to the traditional model of building or renting warehouses. Thus, this article proposes an on-demand approach to the optimization and visualization of the solution to the capacitated warehouse location problem, using genetic algorithms and linear programming with the objective of finding the optimal solu-

tion or approaching it, for a case study applied to Brazilian electronic commerce.

Keywords - capacitated warehouse location problem, on-demand warehousing, genetic algorithms, distributed evolutionary algorithms, linear programming, mixed integer linear programming

I. INTRODUÇÃO

O problema de localização de armazéns capacitados é a base para muitos desafios de otimização combinatória no ramo da logística, onde a demanda total que cada armazém pode satisfazer é limitada. Portanto, consiste em definir um conjunto de armazéns e conectá-los a um conjunto de clientes, com o intuito de minimizar os custos totais de abertura, fechamento e/ou manutenção e conexões para atender os clientes, preservando tanto a satisfação da demanda quanto às restrições de capacidade [10].

Tradicionalmente a abertura, fechamento ou manutenção de armazéns são desafios importantes para as plataformas de comércio eletrônico, empresas de varejo, indústrias, atacadistas e distribuidores, devido às incertezas de demanda dos clientes, a necessidade de construção e/ou aluguel de armazéns, ou mesmo a contratação de operadores logísticos. Logo, a tarefa de definir quantos armazéns, quais são as capacidades, onde abrir e quando fechar tornam-se tarefas importantes para a tomada de decisões [5] [15].

Alternativamente, novos modelos de negócio consideram a armazenagem sob demanda por meio de plataformas com soluções de armazenamento flexível, combinando empresas com armazenamento subutilizado e capacidade de conexão com clientes que precisam de espaço extra ou serviços de conexão. Esses novos modelos de negócios têm vantagens únicas como a granularidade do atendimento e possuem estruturas de custos diferentes em comparação com as formas tradicionais de obtenção de armazéns e serviços de conexão [5] [15].

O comércio eletrônico alterou profundamente as características de demanda e perfis de pedidos. Atualmente, as unidades de movimentação são menores, os locais de entrega são mais dispersos e os clientes esperam entregas rápidas e de baixo custo ou gratuitas. Conseqüentemente, quase metade de todos os varejistas dos EUA oferecem entrega no mesmo dia [13].

Experimentos computacionais ilustram que o poder da armazenagem sob demanda está no aumento do número de locais de armazenamento compartilhado, criando acesso à escala e uso mais eficiente dos armazéns por meio da melhor utilização das suas capacidades [12] [15].

Do ponto de vista da sustentabilidade, o modelo de armazenagem sob demanda pode contribuir para a redução da emissão de carbono, no sentido de ampliar a proximidade do armazém aos consumidores e, evitar ou postergar a construção de armazéns, através de compartilhamento de instalações já existentes [5].

Diante desse contexto, a proposta deste trabalho foi criar um algoritmo genético com algumas restrições que fosse capaz de promover, por meio de otimização e visualização, sob demanda, soluções ou aproximações para o problema de localização de armazéns capacitados.

II. REFERENCIAL TEÓRICO

A. Problema de localização de armazéns capacitados

Os custos de conexão (transporte, manuseio) e os custos fixos específicos do armazém são frequentemente um componente importante do preço (custo) das mercadorias [14].

O problema de localização de armazéns aborda o desafio de localizar um novo conjunto de armazéns de tal forma que a soma desses dois custos seja minimizada [2].

O problema de localização de armazéns capacitados é uma variante do problema de localização de armazéns e leva em consideração as capacidades dos armazéns. Com a inclusão das capacidades, um armazém escolhido que é a fonte de menor custo para um nó de demanda pode não ser capaz de atender a nenhuma demanda de clientes nesse nó [14].

A expansão no número de armazéns pode diminuir os custos variáveis das conexões, por conseguinte, os armazéns ficarão mais próximos dos clientes. Entretanto, ocorre um aumento nos custos fixos dos armazéns em função da ampliação da quantidade dos mesmos [3].

Outra suposição considera que os custos de conexão de armazéns para os clientes são lineares de acordo com o volume transportado ou manuseado, ou seja, não há economia de escala depois de incorrido um custo fixo inicial do armazém [14].

B. Modelos estudados

Uma forma importante de medir a eficácia da localização de um armazém é determinando a distância média percorrida por quem o visita [7].

A seguir alguns modelos que foram estudados a fim de definir qual seria utilizado.

1) *P-Median*

Apresenta uma forma equivalente de medir a eficácia da localização quanto as demandas e ponderar a distância entre os nós de demanda e armazéns pela quantidade de demanda associada e calcular a distância total ponderada de viagem entre demandas e armazéns [11]. Esse problema consiste

em encontrar a localização das instalações de modo a minimizar a distância total de viagem ponderada pela demanda, e entre as demandas e as instalações

2) *K-Median*

É uma variante do problema de localização de instalação não capacitada e visa especificamente estabelecer k instalações, sem considerar custos fixos, ou seja, um determinado número de instalações em pontos possíveis de um gráfico, de forma que a soma das distâncias de cada cliente até a instalação mais próxima seja minimizada [6].

3) *Capacitated K-facility Location Problem*

O problema de localização de armazéns capacitados é localizar vários armazéns para atenderem um conjunto de clientes, a um custo mínimo, onde cada cliente tem uma demanda associada e os armazéns possuem restrições de capacidade de atendimento da demanda. Depois de diversas pesquisas, optou-se por usar este modelo neste trabalho pois ele utiliza a própria rede de fornecedores para serem candidatos a possíveis armazéns. Isso é interessante uma vez que os armazéns que estejam totalmente fora das localidades principais não sejam escolhidos. Assim, a função objetivo que foi utilizada no algoritmo proposto é apresentada na equação (1), onde D é um conjunto de clientes, F é um conjunto de potenciais armazéns, i é o armazém, j é o cliente. Cada armazém i pertence a F e possui capacidade S_i . Cada cliente j possui uma demanda d_j , que deve ser atendida. Estabelecer um armazém i incorrerá em custos fixos de abertura f_i . O transporte X_{ij} unidades, do armazém i para o cliente j , incorrerá em custos de conexão C_{ij} , proporcionais a distância. Assim, a função objetivo que foi utilizada no algoritmo proposto é apresentada a seguir:

$$\sum_{i \in F} \sum_{j \in D} C_{ij} X_{ij} + \sum_{i \in F} f_i y_i \quad (1)$$

$$\sum_{i \in F} X_{ij} = d_j, \forall j \in D, \quad (2)$$

$$\sum_{j \in D} X_{ij} \leq S_i y_i, \forall i \in F, \quad (3)$$

$$\sum_{i \in F} y_i \leq k, \quad (4)$$

$$x_{ij} \geq 0, \forall i \in F, \forall j \in D \quad (5)$$

$$y_i \in \{0, 1\}, \forall i \in F \quad (6)$$

III. METODOLOGIA

Para alcançar os objetivos propostos neste trabalho, a metodologia foi composta por meio do estudo do problema, pré-processamento da base de dados, pela implementação da solução Ótima - *MILP (Mixed integer Linear Programming)* e pela construção do algoritmo genético. A seguir é detalhado sobre eles.

A. Base de Dados utilizada

Este trabalho utilizou a base de dados Kaggle, fundada em 2010 por *Antony Goldbloom* com o objetivo de hospedar competições de *Data Science* e disponibilizar dados sobre diversos assuntos (*datasets*). Utilizaremos o *dataset* “*E-commerce Brazilian Data Set with 100k Orders from 2016 to 2018 - Olist*” [4].

Segundo o *Kaggle*, a base de dados utilizada neste trabalho é um conjunto de dados públicos do comércio eletrônico brasileiro fornecidos pela *Olist Store*, *startup* brasileira do setor *e-commerce* com ênfase no segmento de marketplace. O conjunto de dados contém informações de 100 mil pedidos realizados em diversos mercados no Brasil de 2016 a 2018, além de permitirem a visualização do pedido em várias etapas, desde o status do pedido, preço, desempenho e frete até a geolocalização com as relações dos códigos postais e as coordenadas das entregas. Ademais, os dados em questão são comerciais, reais e foram anonimizados [4].

A *Olist* conecta pequenas empresas no Brasil a canais comerciais que podem vender seus produtos através da Loja *Olist* e enviá-los diretamente aos clientes usando parceiros de logística. O conjunto de dados dos pedidos contém informações dos clientes e suas localizações, ambos utilizados para identificar os clientes e encontrar o local de entrega [4].

B. Solução Ótima – MILP (Mixed Integer Linear Programming)

Com o objetivo de encontrar a solução ótima para o problema de localização de armazéns capacitados, foi utilizado uma biblioteca open source do *Python* denominada *PULP*.

A partir das soluções ótimas encontradas através do *MILP*, entre os diversos cenários avaliados para o problema NP difícil, foram definidos como resultados ótimos aqueles cujo tempo computacional fosse aceitável em termos polinomiais. Desta forma, os resultados obtidos poderiam ser comparados com os resultados futuros da solução aproximada, obtidos na sequência, através do Algoritmo Genético (AG).

O problema de localização de armazéns capacitados é um problema clássico de otimização para determinar quais os armazéns devem abrir ou fechar, de maneira que possam atender uma determinada demanda limitada de pedidos da base *Olist*.

A Figura 1 apresenta quatro modelos para ilustrar como funciona a dinâmica entre os clientes (*customers*), fornecedores (*sellers*) e os possíveis armazéns sugeridos neste trabalho. A Figura 1 (a) é o modelo *Olist* onde o atendimento é feito de forma direta partindo de três fornecedores até seis clientes. A Figura 1 (b) mostra um exemplo de um fornecedor sendo definido com um armazém para atender os seis pedidos dos clientes. Já a Figura 1 (c) é um exemplo de dois fornecedores que foram definidos como armazém e atenderão os seis clientes. Por fim, a Figura 1 (d) é o caso em que três fornecedores são definidos como armazéns para atenderem os seis pedidos dos clientes.

No problema estudado foram definidos como potenciais armazéns todos os fornecedores (*sellers*) estabelecidos em um determinado conjunto amostral de pedidos. O objetivo, portanto, é escolher fornecedores partindo da suposição que dis-

ponham de espaço extra para receber e manusear pedidos adicionais, de forma a contribuir com capacidade expandida de armazenagem através de uma tarifa previamente estabelecida de tal sorte que os custos totais de abertura destes armazéns e os custos de serviços de conexão para atender determinado volume movimentado possam ser minimizados para o conjunto de clientes atendidos.

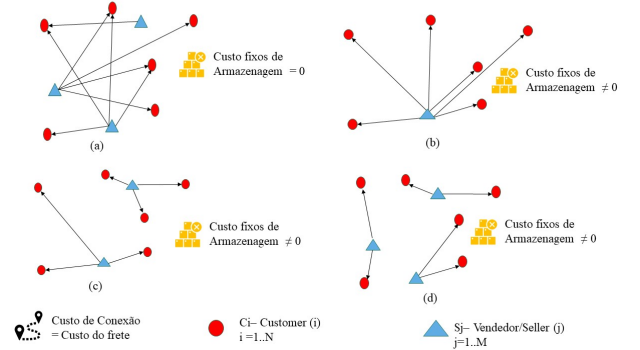


Figura 1: Representação do modelo Olist e do modelo proposto neste trabalho. Fonte: autoria própria.

C. Algoritmo Genético - AG

Algoritmos genéticos são técnicas de otimização e de buscas baseadas no princípio de seleção natural que permite uma população composta de inúmeros indivíduos evoluir sob uma regra definida de seleção. Tal evolução converge para um estado de minimização ou maximização por meio de uma função objetivo, também conhecida por função *fitness* [9]. Eles possuem uma alta capacidade de otimização e podem ser aplicados na solução de sistemas não lineares. As engenharias são exemplos de áreas que fazem uso dos AGs para conseguirem uma primeira aproximação da solução de um problema [9].

Neste trabalho foi proposto um algoritmo genético utilizando a biblioteca *Distributed Evolutionary Algorithms - DEAP*. O *DEAP* é um *framework* desenvolvido na linguagem *Python* que trabalha com computação evolutiva. Ele permite que algoritmos genéticos sejam desenvolvidos utilizando diferentes estruturas para a representação de cromossomos, além de também permitir que sejam criados indivíduos e a manipulação de forma simples dos operadores que compõem um AG [8].

A função *fitness* utilizada no algoritmo desenvolvido é apresentada na equação a seguir.

$$\text{minimize } \sum_{j=1}^m f_j y_j + \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij}$$

Ela é um modelo matemático de otimização que possui a função de minimizar a soma dos custos de ativação das instalações e os custos de conexão. A Figura 2 mostra como é a representação do indivíduo no algoritmo proposto. É possível perceber que cada indivíduo é formado por uma matriz onde as linhas representam os possíveis armazéns e as colunas são compostas pelos clientes. Assim, a matriz é preenchida com 0 e 1, onde 1 significa que aquele armazém entregará o pedido do respectivo cliente. Juntamente com a função *fitness*, também foram criadas quatro restrições que não podem ser violadas ao gerar uma possível solução para o problema. Caso

o resultado de uma solução contenha uma ou mais violações dessas regras será considerada como uma solução inviável.

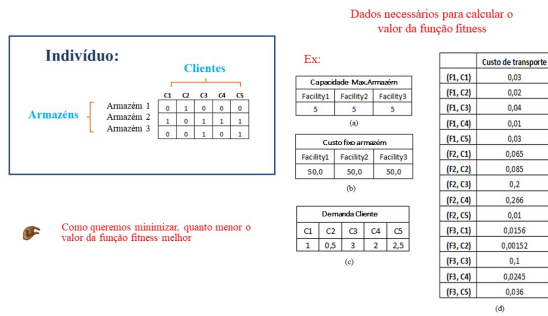


Figura 2: Representação do indivíduo no AG. Fonte: autoria própria. Deste modo, as quatro restrições compõem o AG são:

- Extrapolar a capacidade máxima dos armazéns:** A soma das demandas dos clientes não pode ser maior do que a capacidade de armazenamento de um armazém.
- Não atender a todos os clientes:** Não pode acontecer uma situação em que o pedido de um cliente não seja entregue por nenhum armazém.
- Um cliente sendo atendido por mais de um armazém:** Apenas um armazém pode ser designado para entregar o pedido de um certo cliente, ou seja, não é permitido que dois ou mais armazéns entregarem para o mesmo cliente.
- Os armazéns atendendo mais do que a quantidade de clientes:** Não pode acontecer uma situação em que o total de clientes atendidos seja maior que a quantidade real de clientes.

Com a finalidade de compreender melhor cada uma das restrições, a Figura 3 mostra exemplos de possíveis violações.

O algoritmo foi desenvolvido para que não ocorra a eliminação de uma solução que não cumpra alguma das restrições, mas sim penalizá-las de forma que a solução aproximada não seja uma dessas soluções. Isso é feito por meio de um parâmetro onde se define o valor da penalização. Esse valor será multiplicado pelo somatório das penalizações, ou seja, quanto mais penalidades maiores serão o custo dos armazéns.

O desafio desse tipo de metodologia quando aplicado em algoritmos genéticos é que uma penalização muito pequena pode fazer com que essas soluções pareçam ótimas em comparação com o valor da função *fitness* isolado. Com isso, testes precisam ser feitos a fim de ajustar o fator de peso. Após implementar a função *fitness* e as restrições, o AG foi configurado de modo que foi utilizado a seleção por torneio com tamanho 2, juntamente com o *crossover* de dois pontos e mutação de *bit flip*, ou seja, o gene mutado recebe a negação de seu antigo valor (se era igual a 0 fica igual a 1 e se era igual a 1 fica igual a zero), uma vez que isso é adequado para listas binárias. É importante destacar também que os melhores indivíduos são passados para a próxima geração intactos. Por fim, quando o algoritmo é concluído, são impressos os detalhes da melhor solução que foi encontrada e plotado um gráfico que mostra a função *fitness* e a média obtida nas gerações.

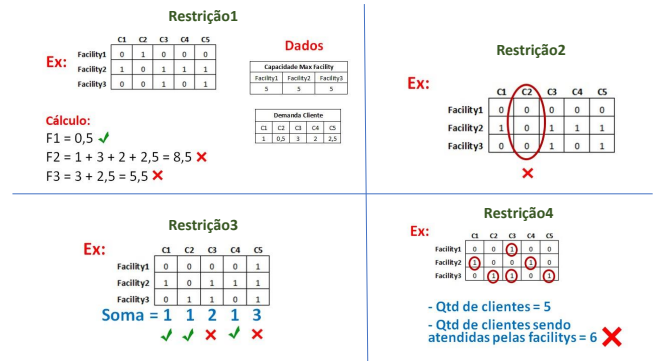


Figura 3: Exemplos de violações das quatro restrições. Fonte: autoria própria.

IV. RESULTADOS E DISCUSSÕES

Este capítulo apresenta a análise dos resultados obtidos com o algoritmo genético criado e os resultados alcançados quando comparado a solução do AG com a solução ótima encontrada pelo MILP.

A. Resultados em algoritmo genético

O algoritmo genético foi executado diversas vezes para encontrar a melhor configuração que se adequasse a base de dados usada. Um dos parâmetros mais importantes a serem ajustados era o valor do fator de penalidade. Isso porque caso ele ficasse com um valor muito baixo, dependendo do valor obtido pela função *fitness*, o algoritmo mostrava como solução, alguma que não respeitava todas as restrições. Além disso, o tamanho da população, a taxa do *crossover*, a taxa de mutação e a quantidade de gerações também foram testadas de modo a encontrar ou se aproximar da solução ótima.

Após diversos testes foi definido que a população = 50, a taxa do *crossover* = 0.9 e a taxa de mutação = 0.1 foram os valores que melhor se adequaram às três bases testadas. Assim, os resultados só variam o valor de fator de penalidade e a quantidade de gerações.

Os primeiros testes realizados foram com uma pequena parte retirada da base de dados completa. Foram selecionados 3 fornecedores e 5 clientes para comporem esses testes. Os 3 fornecedores são informados para o algoritmo como candidatos a armazéns. O algoritmo genético teve que retornar qual dos três armazéns entregariam os produtos aos clientes com o menor custo possível. Como eram poucos dados, o fator de peso foi definido sendo igual a 10.

Gerações	Função Fitness	Resultados				
		Restrições				Custo Final
		R1	R2	R3	R4	
10	5,51	10	0	0	0	15,51
12	3,35	10	0	0	0	13,35
15	6,18	0	0	0	0	6,18
20	6,18	0	0	0	0	6,18
30	6,18	0	0	0	0	6,18

Figura 4: Tabela com os resultados do AG com 5 clientes e 3 possíveis armazéns. Fonte: autoria própria.

A Figura 4 mostra a tabela com os resultados comparativos dessa base com a variação da quantidade de gerações. Quando executada com 10 e 12 gerações o valor da função *fitness* é mais baixo do que os valores de 15, 20 e 30 gerações. Porém,

em ambos os casos a melhor solução viola a primeira restrição e com isso sofre uma penalidade de 10% no valor do custo. Assim, o valor do custo final ultrapassa o de 15 gerações, por exemplo. Outra informação que é possível observar na tabela é que a partir da 15ª geração o algoritmo consegue chegar na solução ótima.

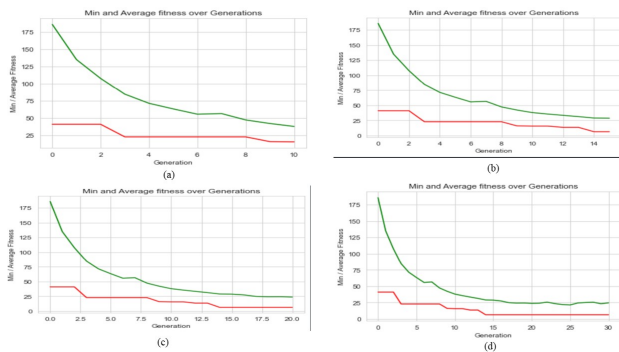


Figura 5: Gráfico com os resultados do AG com 5 clientes e 3 possíveis armazéns. Fonte: autoria própria.

Para melhor visualização, a Figura 5 mostra os resultados por meio dos gráficos gerados. É perceptível que a partir de 14 gerações o algoritmo consegue convergir para o resultado ótimo (linhas vermelhas). A segunda bateria de teste foi feita com mais uma parte selecionada da base de dados. Nesses testes foram usados os pedidos feitos no ano de 2016 até o mês de maio. Assim, o arquivo de teste continha dados de 50 clientes e 30 fornecedores. Cada fornecedor é informado para o algoritmo como um possível armazém. No caso dessa base, o fator de peso foi alterado para 1000 para cada restrição não obedecida, ou seja, o resultado de cada restrição era multiplicado por 1000 e depois era feito um somatório dos valores juntamente com o valor da função *fitness*.

Resultados							
Gerações	Função Fitness	Restrições				Custo Final	QtdeFacilitys
		R1	R2	R3	R4		
1000	10512,76	140000	0	138000	2000	290512,76	30
2000	4334,72	160000	0	0	1600	21934,72	26
3000	5505,12	0	0	0	0	5505,12	25
4000	5459,13	0	0	0	0	5459,13	25
5000	5452,28	0	0	0	0	5452,28	25
10000	5301,69	0	0	0	0	5301,69	23

Figura 6: Tabela com os resultados do AG com 50 clientes e 30 possíveis armazéns. Fonte: autoria própria.

A Figura 6 mostra os melhores resultados dos testes. Por conta do aumento de pedidos com mais instâncias, as gerações também tiveram que aumentar. Na tabela é possível perceber que só após 3000 gerações que o algoritmo consegue gerar soluções que não violem nenhuma restrição

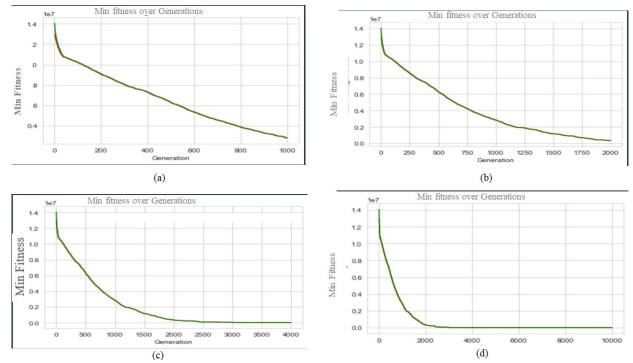


Figura 7: Gráfico com os resultados do AG com 50 clientes e 30 possíveis armazéns. Fonte: autoria própria.

Os gráficos na Figura 7 deixam nítido que a partir de 2500 execuções os resultados começam a se estabilizar. O melhor valor de custo que o algoritmo conseguiu chegar foi de R\$5301,69. Esse valor está um pouco mais alto do que a solução ótima. As Figuras 8 e 9 mostram a tabela e os gráficos do AG para um número maior de pedidos, totalizando 231 pedidos da *Olist*. O aumento da quantidade de pedidos fez com que o algoritmo levasse mais tempo para finalizar as execuções. Mesmo quando testado com 6000 gerações o valor do custo para a melhor solução é alto e ainda assim, ele viola duas restrições. Com isso, os resultados são promissores por se tratar de um projeto em desenvolvimento, e que resultados posteriores serão publicados em projetos futuros.

Resultados						
Gerações	Função Fitness	Restrições				Custo Final
		R1	R2	R3	R4	
1000	807294,38	0	0	9867000	9867000	20541294,4
3000	643326,59	0	0	8472000	8472000	17587326,6
6000	467922,94	0	0	6578000	6578000	13623922,9

Figura 8: Tabela com os resultados do AG com 231 clientes e 101 possíveis armazéns. Fonte: autoria própria.

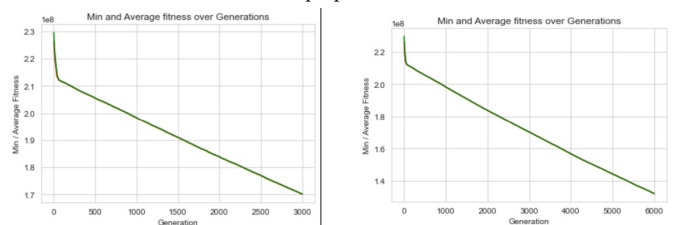


Figura 9: Gráfico com os resultados do AG com 231 clientes e 101 possíveis armazéns. Fonte: autoria própria.

A figura 10 mostra a visualização espacial em duas dimensões da solução real *Olist* (a), a solução ótima do MILP para um armazém (b) e a solução ótima do MILP para 5 armazéns (c), utilizando a biblioteca open source *Folium* em *Python*.

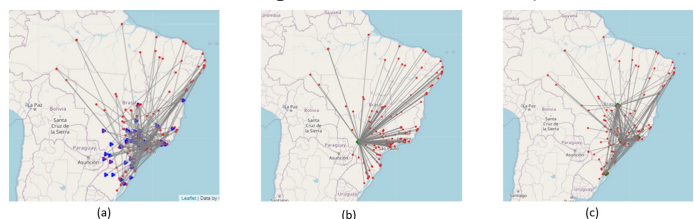


Figura 10: Resultados para do MILP/AG com 230 pedidos. Fonte: autoria própria. Fonte: autoria própria.

A figura 11 ilustra o comparativo de custos entre a solução real *Olist*, com um custo de R\$4.690, a solução ótima encon-

trada pelo MILP por meio de um armazém de R\$6.010 e a solução calculada pelo MILP por meio de 05 armazéns com o menor custo de R\$4376. A solução do AG não foi levada em consideração devido a necessidade de ampliação dos testes e revisão da sua estrutura de funcionamento.

Parameters	Real	Optimal Solution	
# Dynamic Warehouse (on Demand)	0	1	5
# Orders	231	231	231
# Demand (volume m3)	4,33	4,33	4,33
# Capacity (volume m3)	0	5	3
Freight - Seller/Customer	4690	0	0
Storage/Handling - Facility/Customer	0	1000	2500
Freight - Facility/Seller/Customer	0	5010	1876
Objective Function - minimize	4690	6010	4376
Save	0	-1319,75	313,56
Freight Cost seller/Facilities		0 Not Evaluated	Not Evaluated
Total Cost		Not Evaluated	Not Evaluated

Figura 11: Neste caso o resultado do AG não apresentou solução competitiva. Fonte: autoria própria.

V. CONCLUSÃO

Os resultados do AG são promissores pois de forma consistente e proporcional ao número de gerações percebe-se uma evolução nos resultados da função fitness e uma consequente aproximação em relação a solução ótima calculada pelo MILP, não foi possível concluir os limites computacionais do MILP em função da quantidade pequena de pedidos avaliados (5,50 e 230), e como existem 100.000 pedidos na base, seria interessante avaliar um volume massivo de pedidos para identificar o comportamento do mesmo e compará-lo com os resultados da aproximação do AG, para grandes volumes de dados, dado que estamos lidando com um problema NP-difícil. Outro ponto também que poderia contribuir para uma melhor conclusão do estudo, seria avaliar o desempenho utilizando evolução diferencial para que tenhamos uma alternativa de comparação adicional em relação a MILP e AG, o que pode ser feito em estudos futuros. A parte de visualização também pode ser aprimorada levando-se em consideração cenários que utilizem realidade aumentada e realidade virtual em estudos futuros.

REFERÊNCIAS

- [1] Aardal, Karen, et al. "Approximation algorithms for hard capacitated k-facility location problems." *European Journal of Operational Research*, v. 242.2, n.2, p. 358-368, 2015.
- [2] Balinski, Michel Louis. "Integer programming: methods, uses, computations." *Management science*, v. 12, n.3, p. 253-313, 1965.
- [3] Ballou, Ronald H. "Gerenciamento da Cadeia de Suprimentos: Logística Empresarial.", Bookman editora, 2009.
- [4] Kaggle. *Base de dados Kaggle*. Acedido em 24 de Junho de 2021, em: <https://www.kaggle.com/olistbr/brazilian-ecommerce>.
- [5] Ceschia, Sara et al. "The on-demand warehousing problem." *International Journal of Production Research*, p. 1-19, 2022.
- [6] Charikar, Moses et al. "A constant-factor approximation algorithm for the k-median problem." *In: Proceedings of the thirty-first annual ACM symposium on Theory of computing.*, p. 1-10. 1999.
- [7] Church, Richard, and Charles ReVelle. "The maximal covering location problem." *Papers of the regional science association*. Vol. 32. No. 1. Springer-Verlag, 1974.
- [8] DE RAINVILLE, François-Michel et al. "Deap: A python framework for evolutionary algorithms." *In: Proceedings of the 14th annual conference companion on Genetic and evolutionary computation*. p. 85-92. 2012.
- [9] Holland, John H. "Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence." *MIT press*, 1992.
- [10] Melo, Lucas P. et al. "Approximation algorithms for k-level stochastic facility location problems." *Journal of Combinatorial Optimization*, v. 34, n. 1, p. 266-278, 2017.
- [11] Owen, Susan Hesse; Daskin, Mark S. "Strategic facility location: A review." *European journal of operational research*, v. 111, n. 3, p. 423-447, 1998.
- [12] Pazour, Jennifer A.; Unnu, Kaan. "On the unique features and benefits of on-demand distribution models". *15th IMHRC Proceedings - Savannah, Georgia. USA.*, 2018.
- [13] Saleh, K. "The importance of same day delivery—Statistics and trends." Acedido em 30 de Abril de 2020, em: <https://www.invespcro.com/blog/same-day-delivery/>, 2018.
- [14] Silva, Francisco José Ferreira, and D. S. De la Figuera. "A capacitated facility location problem with constrained backloging probabilities." *International journal of production research*, v 45, no. 21. p. 5117-5134, 2007.
- [15] Unnu, Kaan et Pazour, Jennifer. "Evaluating on-demand warehousing via dynamic facility location models." *IIE Transactions*, vol. 54, no 10, p. 988-1003. 2022.