



De 11 a 14 - NOVEMBRO DE 2024

**Inteligência Artificial
na Gestão de Operações:
Limitações e possibilidades**

XXXI SIMPEP
www.simpep.feb.unesp.br

PROPOSTA DE ALGORITMO GENÉTICO PARA A RESOLUÇÃO DE UM MODELO MATEMÁTICO DO PROBLEMA DO TRANSPORTE DE CARTÕES DE CRÉDITO

DENISIE ELLEN DE IOVANNA - denisie.iovanna@ufv.br
UNIVERSIDADE FEDERAL DE VIÇOSA - UFV - RIO PARANAÍBA

LAURA MICHELON NELLESSEN - laura.nellessen@ufv.br
UNIVERSIDADE FEDERAL DE VIÇOSA - UFV - RIO PARANAÍBA

HELEN VITALINE DE CASTRO SANTOS – helen.castro@ufv.br
UNIVERSIDADE FEDERAL DE VIÇOSA - UFV - RIO PARANAÍBA

LAÍS GONÇALVES VIANA – lais.viana@ufv.br
UNIVERSIDADE FEDERAL DE VIÇOSA - UFV - RIO PARANAÍBA

THIAGO HENRIQUE NOGUEIRA – thiagoh.nogueira@ufv.br
UNIVERSIDADE FEDERAL DE VIÇOSA - UFV - RIO PARANAÍBA

ÁREA: 3. PESQUISA OPERACIONAL

SUBÁREA: 3.1. MODELAGEM, SIMULAÇÃO E OTIMIZAÇÃO

RESUMO: ESTE ARTIGO PROPÕE UMA ABORDAGEM PARA MODELAR A DISTRIBUIÇÃO DE CARTÕES DE CRÉDITO UTILIZANDO ALGORITMOS GENÉTICOS (AGS). O MODELO UTILIZADO É BASEADO EM UM ESTUDO ANTERIOR, O QUAL UTILIZOU UM SOFTWARE ESPECÍFICO PARA A DISTRIBUIÇÃO DE CARTÕES. NESSA ABORDAGEM, BUSCA-SE DESENVOLVER UMA METAHEURÍSTICA QUE OFEREÇA RESULTADOS MAIS EFICIENTES EM TERMOS DE TEMPO E QUE POSSA SER APLICADA DE FORMA MAIS AMPLA. A METODOLOGIA ENVOLVE A DEFINIÇÃO DE PARÂMETROS, COMO TIPO DE CARTÃO, SERVIÇO, NÚMERO DE GRÁFICAS E REGIÕES, OS QUAIS SÃO OTIMIZADOS ATRAVÉS DE AGS. ESTE TRABALHO CONTRIBUI PARA A LITERATURA AO APRESENTAR UMA ALTERNATIVA VIÁVEL E EFICAZ PARA A DISTRIBUIÇÃO DE CARTÕES DE CRÉDITO, ALÉM DE DESTACAR O POTENCIAL DOS AGS EM PROBLEMAS DE OTIMIZAÇÃO COMPLEXOS.

PALAVRAS-CHAVE: OTIMIZAÇÃO, MODELO MATEMÁTICO, ALGORITMO GENÉTICO, PESQUISA OPERACIONAL, LOGÍSTICA, CARTÃO DE CRÉDITO.

1. INTRODUÇÃO

Nos últimos anos, as diversas formas de pagamento vivenciaram um crescimento exponencial, passando por diversas fases e impulsionando inovações nas transações através da praticidade. Dados do Banco Central do Brasil (2023) indicam que até o fim desse mesmo ano, os cartões foram o meio de pagamento com maior participação em termos de quantidade de transação, o que atesta a relevância do mesmo na rotina das pessoas, demonstrando a necessidade da aquisição do cartão de maneira eficiente. Dessa forma, fica clara a importância de fornecer aos clientes experiências logísticas que diminuam cada vez mais essa espera, especialmente no contexto acelerado das rotinas atuais, onde a internet e a possibilidade de transações online criaram compradores focados em velocidades extremamente rápidas e com expectativas elevadas (Pedreira, 2016).

Assim, não apenas para os clientes, a correta distribuição de rotas também tem um grande impacto na economia da empresa, uma vez que uma logística pouco eficaz pode ter como consequência perdas e corrimento de custos extras de transporte e estocagem (Hausman, Lee e Subramanian, 2013). Além disso, por contar com um grande catálogo de produtos produzidos em diferentes gráficas, espalhadas pelo território e distribuídas por operadores logísticos com serviços variantes, a cadeia de suprimentos de cartões possui uma certa complexidade, pois conciliar essa grande variabilidade de parâmetros pode se tornar um desafio. Dessa forma, a construção de um modelo matemático e sua solução em tempo ótimo pode ser extremamente útil para auxiliar na construção dessas rotas, por comparar diferentes combinações e trazer tempos e lucro em um tempo muito menor e com possibilidade menor de erro se comparado com uma pessoa (Pinto et al., 2023).

Dessa forma, o seguinte texto busca apresentar uma proposta de melhoria para o modelo matemático de Pinto et al. (2023), o qual tem como objetivo o cálculo das rotas para a distribuição de cartões de crédito com o intuito de maximizar a receita. Dentro deste artigo é citada a dificuldade dos autores de trabalharem com situações mais realísticas, devido ao tempo de execução que torna-se inviável com uma quantidade maior de parâmetros. Assim, com o intuito de aumentar as possibilidades de testes, este trabalho tem o objetivo de propor uma metaheurística baseada em algoritmos genéticos, que consiste na seleção probabilística de algumas soluções que serão eliminadas ou evoluídas de acordo com a qualidade da solução (Lopes, 1995) em substituição do método exato Branch and Cut. Sendo assim, essa abordagem busca expandir a dimensão do modelo trabalhando com soluções viáveis encontradas em

tempos computacionais razoáveis, de modo a trabalhar com parâmetros cada vez mais amplos e adaptados à necessidade de empresas reais.

Por fim, o artigo está dividido em 5 partes. A primeira é a introdução, na qual busca-se explicitar o contexto e objetivo do artigo, de forma a clarificar os pontos a serem abordados; em seguida, o artigo possui um referencial teórico que discute sobre a natureza do problema e do tema, o que inclui: a discussão sobre o problema do transporte aplicado na distribuição de cartões de crédito, a utilização de métodos exatos e heurísticas para resolução de problemas, assim como um aprofundamento na teoria sobre os algoritmos genéticos e sua utilização nesse tipo de problema; também será trazido um tópico sobre resultados e discussões, no qual serão feitas todas as análises e comparações com o artigo original; o texto será finalizado com uma conclusão que terá como o objetivo entender os aprendizados.

2. REFERENCIAL TEÓRICO

2.1 O problema do transporte de cartões de crédito

O problema do transporte, definido como a atribuição de quantidades de certa mercadoria, transportadas de determinado local para um destino, obedecendo a limitações de ofertas e demanda, tem sido amplamente estudado e possui grande relevância em diversas áreas, especialmente na logística e na administração de cadeias de suprimentos (Oliveira, 2016). A aplicação deste problema é muito vasta, podendo abranger desde a distribuição de produtos e serviços até a logística de transporte de produtos bancários como os cartões de crédito, no qual o problema do transporte é particularmente desafiador devido às necessidades específicas de segurança e rastreamento. Este processo de distribuição é realizado por meio de operadores logísticos terceirizados que fazem o recolhimento dos cartões nas várias gráficas e os direcionam diretamente para as regiões onde os clientes estão localizados (Pinto et al., 2023). Desse modo, a complexidade do problema aumenta devido à necessidade de sincronização entre a produção dos cartões, a coleta pelos operadores logísticos e a entrega final aos clientes.

Assim, a modelagem matemática do problema do transporte de cartões de crédito pode ser feita baseando-se na formulação do Problema do Transporte clássico, que inclui variáveis de decisão, uma função objetivo e um conjunto de restrições (Pimentel, 2020). No caso do transporte dos cartões, os parâmetros devem ser modificados para englobar as dificuldades de se ter diferentes gráficas (fornecedores), clientes (consumidores) e janelas de tempo para as entregas. Assim, as variáveis de decisão representam a quantidade de cartões produzidos e

estocados em cada gráfica, a quantidade enviada por cada operador logístico e a quantidade atendida de cada região; enquanto isso, a função objetivo busca maximizar a diferença entre a receita arrecadada com a entrega dos cartões e os custos operacionais envolvidos nos processos de produção e distribuição; e por fim as restrições garantem que a capacidade produtiva das gráficas, os estoques, a distribuição, e os atendimentos de demandas sejam respeitados (Pinto et al., 2023).

2.2 Heurísticas e métodos exatos

A resolução de problemas de otimização pode se dar de duas formas, a utilização de métodos exatos que buscam resolver problemas na otimalidade, mas que, em contrapartida, requerem uma capacidade computacional muito maior, e as heurísticas que sacrificam a solução na otimalidade para conseguir resolver problemas de maneira simplificada e que definirá o limite inferior ou superior para solução a depender do tipo de problema (Cintra et al., 2008). A escolha entre essas duas formas de resolução depende não apenas da natureza do problema, mas também da disponibilidade de recursos computacionais e da escalabilidade do problema, tendo uma influência direta na complexibilidade e tempo de execução ao tentar resolvê-lo (El-Sherbeny, 2010). A crescente complexidade dos problemas de otimização na prática tem impulsionado o desenvolvimento de abordagens híbridas, que combinam heurísticas e métodos exatos para aproveitar as vantagens de ambos (Muthuraman e Venkatesan, 2017).

Plous (1993) considera as heurísticas como um conjunto de regras que auxilia no julgamento em tarefas decisórias de incerteza graças à redução de tempo e esforços empreendidos. Dessa forma, as heurísticas tornam-se fortes candidatas para a resolução de diversos problemas do dia-a-dia por reduzir a complexidade de tarefas e chegar em resultados com simples julgamentos (Tonetto et al., 2006). Além disso, é importante reconhecer que as heurísticas possuem certas limitações devido essa busca pela redução da complexidade e que acarreta, muitas vezes, na não garantia de encontrar soluções corretas ou ótimas, o que faz com que a escolha da heurística tenha que ser bem estudada e fundamentada (Abel, 2003). As heurísticas podem ser classificadas em heurísticas de construção, que constroem uma solução inicial passo a passo, heurísticas de melhoria, que utilizam técnicas para melhorar uma solução já existente e, por fim, as metaheurísticas que combinam diferentes heurísticas para a resolução de problemas mais complexos (Oliveira e Carravilla, 2009).

2.3 Algoritmos genéticos

O algoritmo genético surgiu na tentativa de incorporar aspectos da evolução nos sistemas computacionais e hoje são utilizados na resolução de diversos problemas de pesquisa operacional e otimização (Das e Suganthan, 2011). Esse algoritmo consiste na seleção de “cromossomos” que seria um conjunto inicial de soluções, a ser gerado aleatoriamente ou não, a ser cruzados na intenção de formar novos indivíduos que darão origem a uma nova geração e que são formados a partir da junção dos genes dos dois pais, ou seja, a partir da combinação de duas soluções anteriores (Holland, 1992). A continuidade da existência desses indivíduos dependerá da seleção natural, após a formação de uma nova geração os indivíduos mais aptos sobrevivem e os outros desaparecem, em termos do algoritmo, alguns indivíduos serão selecionados a continuar no conjunto solução a partir de um critério previamente definido, que avaliará a qualidade das soluções, as piores soluções desapareceram (Mitchell, 1995). Ainda de acordo com o mesmo autor, após a seleção o ciclo se inicia e novamente há um cruzamento entre os pais restantes, seguido por uma seleção, esse processo continua até que o critério de parada seja atingido.

Devido à grande versatilidade de aplicações e suas vantagens, os algoritmos genéticos são amplamente utilizados em diversas áreas como engenharia, ciência da computação, economia, entre outros, com aplicações cada vez mais abrangentes (Yang, 2010). Em problemas logísticos e da cadeia de suprimento, por exemplo, a aplicação de metaheurísticas se dá em diferentes contextos como no planejamento de rotas, produção, cadeia de produção e sistemas logísticos, já na engenharia, pode-se ver sua participação na otimização de sistemas complexos e processos industriais (Talbi, 2009). No texto de Deb e Jain (2014), os autores destacam algumas dessas vantagens, que incluem uma resolução relativamente eficaz de problemas complexos se comparados com outras heurísticas, combinado com uma ampla exploração do espaço de busca e a flexibilidade e facilidade de implementação.

2.4 Aplicação de algoritmos genéticos no problema do transporte de cartões de crédito

Os AGs têm mostrado vantagens em diversos problemas de otimização quando comparados com os métodos tradicionais, principalmente devido à capacidade de explorar grandes espaços de busca e de adaptar-se a diferentes cenários com apenas pequenas modificações (Gómez et al., 2014). A aplicação desses algoritmos ao Problema do Transporte envolve a adaptação de seus elementos básicos para representar adequadamente o problema.

Isso equivale a dizer que deve-se definir cromossomos que representem as possíveis soluções, uma função de aptidão que avalie a qualidade dessas soluções, e estratégias específicas de seleção, cruzamento e mutação de forma a respeitar todas as particularidades do problema (Rocha, 2023). Exemplificando, um gene é representado por um ponto cartesiano e um cromossomo por uma rota de pontos; a população é um conjunto de rotas e a função de adaptação calcula a distância euclidiana de uma rota, atribuindo uma pontuação inversamente proporcional à distância; pais são duas rotas combinadas para formar descendentes sem repetição de pontos, enquanto mutação troca posições entre pontos; e a seleção elitista escolhe as rotas de menor distância proporcional à aptidão (Santos Neto, 2023).

Dessa forma, para retratar o contexto específico da distribuição de cartões de crédito, surgem alguns desafios adicionais com relação à segurança, à eficiência e ao custo das operações, principalmente pelo alto valor da carga e pelos riscos de furto e extravio (De Carvalho, Soliani e De Freitas, 2022). Nesse contexto, a otimização é crucial para reduzir custos operacionais e aumentar a eficiência da distribuição e ainda garantir soluções eficientes e seguras para o transporte de itens de alto valor como cartões de crédito (Lopes, Galvão e Fogaça, 2015). Assim, a comparação de desempenho entre os métodos exatos tradicionais de resolução do Problema do Transporte e os AGs é essencial para validar a eficácia das abordagens e poder determinar qual é mais vantajoso com relação à velocidade e qualidade das soluções. Nesse sentido, sabe-se que devem ser apresentados resultados empíricos para ilustrar os ganhos de desempenho e a viabilidade prática do algoritmo genético, o que será desenvolvido nas próximas seções deste trabalho.

3. MODELO E MÉTODO

3.1 Modelo matemático

Pinto, Amaral e Santos (2023) formularam o problema de distribuição de cartões como um modelo matemático, de modo a maximizar o lucro obtido por meio da receita dos vários tipos de cartão, considerando as perdas de receita de entregas adiadas e custos de produção, armazenamento e distribuição. Os autores também estimaram no modelo que as gráficas possuem limitações de produção, que os cartões são distribuídos utilizando diversos operadores logísticos e serviços e que os produtos são encaminhados diretamente das gráficas para os clientes (Pinto, Amaral e Santos, 2023). Assim, a seguir pode-se visualizar uma síntese da formulação analisada neste artigo, em que se buscou manter a proposta dos autores com o

mínimo de alteração possível para permitir a comparação de resultados; para informações mais detalhadas sobre o modelo e os resultados obtidos pela modelagem linear recomenda-se ler o artigo mencionado.

$$\begin{aligned}
 & (\max) \sum_{c \in NC} \sum_{m \in NGm} \sum_{t \in T} R_{cs} v_{cmst} \\
 & - \sum_{c \in NC} \sum_{g \in NG} \sum_{m \in NGm} \sum_{s \in NS} \sum_{o \in NO} \sum_{t \in T} \sum_{t' \in T} \alpha_{cgmostt'} (t - t') R_{cs} Z \\
 & - \sum_{c \in NC} \sum_{g \in NG} \sum_{t \in T} P_{cg} x_{cgt} - \sum_{c \in NC} \sum_{g \in NG} \sum_{t \in T} H_{cg} I_{gt} \\
 & - \sum_{g \in NG} \sum_{t \in T} \sum_{m \in NGm} \sum_{s \in NS} \sum_{o \in NO} W_{gtms} (\sum_{c \in NC} \alpha_{cgmostt'})
 \end{aligned}$$

S.a.:

- (1) $\sum_{c \in NC} x_{cgt} \leq Cap_g, \forall g \in NG, t \in T$
- (2) $x_{cgt} \leq BigM_{ycg}, \forall c \in NC, g \in NG$
- (3) $x_{cgt} \geq E_{cgycg}, \forall c \in NC, g \in NG$
- (4) $I_{cg} = I_{cg} + x_{cgt} - \sum_{m \in NGm} \sum_{o \in NO} \sum_{s \in NS} \sum_{t' = t - rc} \alpha_{cgmos}, \forall c \in NC, g \in NG, t \in T \setminus \{1\}$
- (5) $I_{cg} = x_{cg1} - \sum_{m \in NGm} \sum_{o \in NO} \sum_{s \in NS} \alpha_{cgmos}, \forall c \in NC, g \in NG$
- (6) $\sum_{o \in NO} \lambda_{omt} \geq F_m, \forall m \in NGm, t \in T$
- (7) $\sum_{t' \in T} \sum_{c \in NC} \sum_{g \in NG} \sum_{s \in NS} \alpha_{cgmos} \geq V O_m \lambda_{omt}, \forall m \in NR, o \in NO, t \in T$
- (8) $Q_{cmt} \leq \sum_{s \in NS} v_{cmst} \leq D_{cmt}, \forall c \in NC, m \in NR, t \in T$
- (9) $\sum_{g \in NG} \sum_{o \in NO} \sum_{s \in NS} \sum_{t=1} \sum_{t'=1} \alpha_{cgmos} \leq \sum_{s \in NS} \sum_{t=1} v_{cmst} \forall c \in NC, m \in NR, u \in T$
- (10) $\alpha_{cgmos} = 0, c \in NC, g \in NG, m \in NR, o \in NO, s \in NS, t \in T, t' \in \{0, \dots, t - rc - 1\} \cup \{t + 1, \dots, T\}$
- (11) $\sum_{g \in NG} \sum_{o \in NO} \sum_{s \in NS} \sum_{t=1} \sum_{t'=1} \alpha_{cgmos} \geq \sum_{s \in NS} \sum_{t=1} v_{cmst}, \forall c \in NC, m \in NR, u \in T$
- (12) $\sum_{g \in NG} \sum_{o \in NO} \sum_{s \in NS} \sum_{t \in T} \alpha_{cgmos} = \sum_{s \in NS} v_{cmst}, \forall c \in NC, m \in NR, t' \in T$
- (13) $\sum_{g \in NG} \sum_{o \in NO} \sum_{s \in NS} \sum_{t=1} \alpha_{cgmos} \geq K \sum_{s \in NS} v_{cmst}, \forall c \in NC, m \in NR, t' \in T$
- (14) $x_{cgt} \geq 0, I_{cg} \geq 0, y_{cgt} \geq 0, \forall c \in NC, g \in NG, t, t' \in T$
- (15) $\alpha_{cgmostt'} \geq 0 \forall c \in NC, g \in NG, m \in NR, o \in NO, s \in NS, t, t' \in T$
- (16) $\lambda_{omt} \in \{0, 1\}, y_{cgt} \in \{0, 1\}, \forall c \in NC, g \in NG, m \in NR, o \in NO, t \in T$

3.2 Método de resolução

Em seu trabalho, Pinto, Amaral e Santos realizaram os testes pelo método branch and cut de um resolvidor pronto, em que obtiveram bons resultados que permitiram comprovar a viabilidade do modelo matemático. Entretanto, os autores entenderam que haveria a necessidade de melhorar a velocidade de resolução e aumentar o conjunto de parâmetros em uma continuação de sua pesquisa, para que os resultados abrangessem melhor a realidade de uma empresa que quisesse usá-los. Desse modo, na sequência, este trabalho irá apresentar uma metaheurística de algoritmo genético com exatamente esse intuito de resolver o modelo proposto com maior quantidade de instâncias e em um tempo operacional polinomial. Assim, abaixo é apresentado o pseudocódigo para o algoritmo genético e, em seguida, uma breve explicação de seu funcionamento.

Pseudocódigo

1. Inicialização:

- Chame as bibliotecas necessárias
- Defina a população inicial de indivíduos (representando soluções)
- Determine os parâmetros do algoritmo

2. Avaliação:

- Inicialização de relógio para avaliação da eficiência
- Calcule a função objetivo para cada indivíduo (receita - custo)
- A função de custo inclui:
 - Custos de produção de cartões de crédito.
 - Custos de transporte entre locais.
 - Custo de armazenamento

3. Seleção:

- Definição dos parâmetros do algoritmo genético
 - Selecione os melhores indivíduos com base em sua função objetivo utilização do Torneio

4. Reprodução:

- Crie novos indivíduos (filhos) a partir dos indivíduos selecionados.
- Utilização do crossover para combinar informações genéticas dos pais

5. Mutação:

- Adição de algumas variações aleatórias nas soluções criadas

6. Substituição:

- Substitua parte da população antiga pelos novos indivíduos gerados.
- Isso mantém a diversidade e permite a evolução contínua.

7. Critério de Parada:

- Verifique se o critério de parada foi atingido (Número máximo de gerações)
- Se o parâmetro for atingido, vá para o passo 7
- Senão, volte para o passo 2

8. Resultado:

- O melhor indivíduo encontrado representa a solução otimizada para o problema de redução de custos.

O algoritmo é iniciado com a inclusão das bibliotecas que serão utilizadas. A primeira é a numpy, uma biblioteca da linguagem Python utilizada para gerar valores aleatórios e inseri-los em listas randômicas que vão colocar em sequência as instâncias e demais dados necessários (dias, cartões, gráficas, regiões, etc). A próxima, a biblioteca random, será responsável por embaralhar o conteúdo dessas listas, combinando-as em modo aleatório. Em seguida é incluída a biblioteca time que será utilizada para a cronometragem do tempo de execução, um valor importante para comparação de soluções rápidas. Por fim, a última biblioteca é a deap, que importa algumas partes da estrutura do algoritmo genético (base, creator, tools e algorithms) e simplifica significativamente sua construção; assim, apenas será necessário atribuir os valores dos parâmetros para a execução do algoritmo, uma vez que as funções responsáveis pela seleção, reprodução e mutação já estão previamente definidas.

Após essa inicialização, são definidos os parâmetros de custo, receita, capacidade e demanda, valores iniciais gerados a partir das bibliotecas numpy e random, na qual é definido o intervalo de valores e o tamanho dos vetores. Então são criadas as variáveis que guardarão os resultados das gerações que serão utilizadas pelo algoritmo genético.

Em seguida, é definida a função evaluate, que é responsável por determinar a aptidão de cada indivíduo na população. O funcionamento dela se dá através do cálculo da função objetivo do problema, ou seja, o lucro total utilizando determinadas instâncias, e verifica se o nível de serviço é satisfatório; se não, o indivíduo é penalizado de forma a influenciar na seleção para a reprodução da próxima geração. A "função objetivo", como o nome sugere, representa nosso objetivo, ou seja, obter o maior lucro possível, melhorando os critérios de produção, armazenamento e distribuição de cartões de crédito, principalmente reduzindo os custos. No trabalho de referência, infelizmente, não são fornecidos dados reais sobre as atividades em questão, pois são estritamente confidenciais para os bancos fornecedores, mas são usados dados aleatórios em intervalos de medidas semelhantes à realidade. Por esse motivo, também geramos nossos próprios dados usando a biblioteca random, com a restrição dos mesmos intervalos de medidas do trabalho de referência, inserindo-os em listas dinâmicas. Para obter combinações de casos mais imprevisíveis, então misturamos os dados de cada lista individualmente.

O coração do programa é a biblioteca deap, que implementa todas as funcionalidades e sequências do "algoritmo genético". É uma biblioteca muito complexa, permitindo o uso de configurações diferentes para múltiplos usos diferentes do "algoritmo genético". Em nosso caso, usamos apenas as partes "base", "creator", "tools", "algorithms" e, desta última, a função "eaMuPlusLambda", que executa todas as etapas necessárias para a execução do algoritmo genético. É possível controlar os critérios de evolução por meio dos parâmetros: 1) population, que é uma lista que conterá os indivíduos escolhidos para realizar a "função objetivo"; 2) toolbox, uma classe que contém, entre outras coisas, os operadores de evolução dos vários indivíduos; 3) mu, o número de indivíduos selecionados para a próxima geração; 4) lambda, o número de filhos criados em cada geração; 5) cxpb, a probabilidade de um descendente ser gerado por cruzamento; 6) mutpb, a probabilidade de um filho ser gerado por mutação em relação ao anterior; e 7) ngen, o número de gerações desejadas.

Neste ponto, é realizado um loop aninhado para associar a cada indivíduo de cada geração todas as combinações possíveis de cartão de crédito, design, região, tipos de serviços de distribuição e dias da semana. Com base nessas combinações, o lucro possível é calculado, então se a capacidade de design puder atender à solicitação, o lucro é aceito e passamos a verificar o tipo de serviço de distribuição solicitado e se o tempo previsto para a entrega for respeitado; se sim, o lucro é alcançado, caso contrário, haverá uma penalidade, até que o lucro seja zerado no caso de entrega urgente não realizada.

Por fim, como o objetivo do programa é demonstrar que o tempo de execução do "algoritmo genético" é inferior ao do "branch and cut" a cada geração de indivíduo e ativação da "função objetivo", um cronômetro é iniciado para calcular o tempo de computação; o cronômetro é parado quando os resultados são retornados. Um cronômetro também é ativado para calcular o tempo total de execução do programa. No final do programa, os dados relativos ao melhor e pior lucro e às médias de lucro e tempo de computação são impressos.

4. RESULTADOS E DISCUSSÕES

4.1 Observações sobre a geração de resultados

O objetivo do trabalho foi propor uma metaheurística com algoritmo genético em substituição ao método exato do branch and cut utilizado originalmente no modelo matemático utilizado, e comparar os resultados para provar que o primeiro é um método mais rápido e suficientemente adequado para alcançar resultados. Desse modo, uma das principais

características que diferenciam os métodos é a forma pela qual o problema é desenvolvido, o que inclui como as soluções são geradas, os cortes são feitos e a iterações dentro do código. Devido a esse fato, é crucial entender que a distribuição dos resultados é diferente e deve-se apegar ao objetivo final, ou seja, definir a eficiência do método proposto em relação ao tempo de execução de código e otimalidade das soluções.

Além disso, é importante ressaltar que a comparação entre o artigo analisado e o presente trabalho não pode ser feita de forma totalmente direta, uma vez que os dados utilizados em Pinto, Amaral e Santos não são reais e foram gerados de maneira aleatória dentro de uma faixa pré-estabelecida, assim como neste artigo. Entretanto, as gerações foram construídas buscando seguir a estruturação do código original, ou seja, em relação ao número de gerações e de indivíduos que viriam a ser utilizados, o que permite a criação de inferências ao comparar tempos e oscilações. Assim, para a definição das características dos indivíduos foi definido aleatoriamente um valor booleano (0 ou 1) que seria multiplicado por cada um dos genes que estão sendo definidos de acordo com o número máximo de cartões, gráficas, regiões e dias, para que as características de cada um fossem diferentes e aleatórias. Isso significa que o valor do número de cartões, gráficas, regiões e dias variaram de 1 a n como pode ser observado na tabela 1 abaixo.

TABELA 1 - Valores para a distribuição aleatória dos parâmetros

Genes	Cartões	Gráficas	Regiões	Dias
n	15	15	15	30

Fonte: elaborado pelas autoras (2024).

Seguindo a estrutura do algoritmo genético, o código tem como condição de parada o número de gerações. Dessa forma, ele é executado para 30 gerações que geram 100 indivíduos cada, resultando em 3000 instâncias analisadas no total. Assim, como um dos principais fatores para a decisão de viabilidade neste estudo é o tempo, o processador e sistema operacional da máquina utilizada para os testes é um fator de variação considerável na geração dos resultados, principalmente nesse caso em que há um volume de dados muito grande. Desse modo, com o intuito de gerar uma comparação mais justa, o código foi executado em duas máquinas com características computacionais diferentes: a primeira possui um processador Intel Core i5-10400F (12MB de cache, velocidade de clock de até 4.30 GHz no modo turbo) com 12 threads, 16GB de RAM HyperX Fury DDR4 2666MHz XMP e um disco rígido Crucial BX500 de 240GB e a segunda máquina possui um processador Intel(R) Core(TM) i5-5200U, com 8 GB

de memória RAM e sistema operacional Windows 10 Home Single Language, que se aproxima mais ao utilizado no artigo original.

4.2 Resultados obtidos e discussão

Após a execução do código em ambas as máquinas, cada uma das 30 gerações retornou o valor de Função Objetivo (FO) e o tempo de execução daquela geração. Desses valores, anotou-se o melhor e o pior valor de FO, assim como para o tempo total de execução do algoritmo e o tempo médio das gerações, ao fim, realizou-se uma média de FO. Ao observar os resultados para ambas as máquinas, apesar da diferença entre os tempos de execução, é possível perceber um comportamento em comum em relação aos resultados devolvidos, demonstrando que mesmo com diferentes máquinas que geram resultados em tempos distintos, os resultados tendem a convergir para um lucro similar. Um resumo prático dos resultados gerados com a execução do código nas duas máquinas podem ser observados na tabela abaixo.

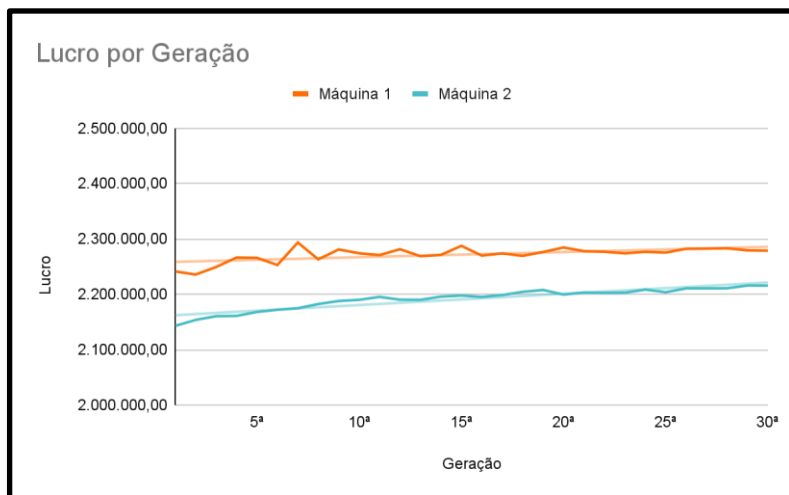
TABELA 2 - Resultados

Métricas	Artigo Base	Máquina 1	Máquina 2
FO Média	R\$ 480.040,7	R\$ 2.201.229,69	R\$ 2.191.672,47
Melhor FO	R\$ 2.141.224,39	R\$ 2.293.494,01	R\$ 2.301.712,10
Pior FO	R\$ 47.723,67	R\$ 2.101.535,91	R\$ 2.087.753,41
Tempo total (s)	1898,07	1317,16	2699,71

Fonte: elaborado pelas autoras (2024).

A plotagem dos valores de FO obtidos nas duas máquinas gerou um gráfico de linha que ilustra o comportamento dos valores ao longo das iterações e sua tendência (Figura 1). Os dados do gráfico mostram uma tendência de crescimento com uma diminuição contínua da variabilidade, confirmando que o algoritmo funciona conforme esperado. Assim, podemos concluir que o algoritmo parte de um conjunto de soluções iniciais e evolui em direção a resultados melhores, resultando em lucros cada vez mais constantes.

FIGURA 1 - Lucro por Geração



Fonte: elaborado pelas autoras (2024).

Ao comparar esses resultados com os do artigo original, percebe-se que, embora o tempo de execução na Máquina 2 seja maior, isso se deve à característica do método de execução. No original, ao atingir um tempo operacional superior a 300 segundos, a operação é interrompida. Portanto, o tempo total de execução não reflete a realidade completa, ao contrário do que ocorre na nossa implementação, onde todas as instâncias foram geradas sem interrupções. Além disso, com este método, é possível convergir para um bom resultado mais rapidamente, atingindo-se o melhor valor de FO nas primeiras iterações.

Por fim, é importante ressaltar que a viabilidade das heurísticas está atrelada ao tempo de execução, assim, ao estabelecer paralelos entre elas e os métodos exatos, sempre busca-se ressaltar a eficiência. Dessa forma, durante a construção do algoritmo surgiram algumas dificuldades que seriam decisivas para o sucesso do código. A primeira se tornou uma adaptação: no artigo estudado, os autores fazem uso de um solver que os permite resolver problemas em tempo competitivo e com facilidade; devido ao tipo de input que o programa permite, nele insere-se o modelo matemático e grande parte é resolvida pelo solver. Nesse sentido, surge um desafio em relação ao tempo, pois o código em Python necessitava ser construído de forma que a sua execução fosse tão eficiente quanto a original sem prejudicar muito a qualidade das soluções. Assim, a utilização da biblioteca *deap* facilitou essa implementação e o código foi construído e revisado com o intuito de evitar ao máximo redundâncias.

Por outro lado, uma segunda dificuldade na realização desse projeto foi o tamanho do modelo matemático, pois sua função objetivo possui mais de 20 somatórias e 16 restrições, o que traz certa complexidade na elaboração das funções, sempre com uma preocupação voltada para o tempo de execução. Dentro disso, para assegurar a correta execução do código foi

necessário validar alguns pontos, como os parâmetros que estão sendo passados e avaliar se o código estava se prendendo a soluções subótimas, por isso a inserção da taxa de mutação foi importante para conservar a diversidade genética.

5. CONCLUSÃO

Neste trabalho foi apresentado o modelo matemático criado por Pinto, Amaral e Santos (2023) com o objetivo de maximizar o lucro obtido por meio da receita de determinado número de tipos de cartão, quantidade de regiões e gráficas abrangidas. O objetivo principal deste trabalho foi propor um algoritmo genético para a resolução da modelagem e comparar com os resultados obtidos para a mesma modelagem por um método exato. Assim, foi desenvolvido um código em linguagem Python que forneceu soluções para 3000 instâncias em um tempo médio de 0,44s por instância, obtendo um resultado de Função Objetivo razoavelmente ótima.

Desse modo, foi possível concluir que heurísticas para resolução de problemas matemáticos complexos como distribuição de recursos têm capacidade de chegar em resultados relevantes quando comparados com métodos exatos, pois propicia resultados parecidos em um tempo computacional extremamente menor. Além disso, pôde-se notar que o algoritmo genético para resolução do problema de distribuição foi uma ótima escolha, principalmente pela facilidade das bibliotecas pré-prontas da linguagem python. Por último, também foi possível demonstrar que a capacidade de processador é uma variável importante para a resolução de programas extensos, pois processadores mais potentes reduzem significativamente o tempo de execução do código.

Em conclusão, a utilização de algoritmos genéticos demonstrou-se uma abordagem eficaz na otimização de problemas complexos de distribuição de recursos, fornecendo uma alternativa viável aos métodos exatos em relação à sua eficiência computacional. Este estudo reafirmou a importância de investir em hardwares adequados para a execução de algoritmos complexos, além de destacar o valor das ferramentas disponíveis em Python para o desenvolvimento de soluções rápidas e eficazes. Esses resultados incentivam a exploração contínua de heurísticas e metaheurísticas em diversos campos da otimização, apontando para um futuro promissor na solução de problemas matemáticos através de métodos inovadores e computacionalmente eficientes.

6. REFERÊNCIAS BIBLIOGRÁFICAS

ABEL, Charles. (2003). **Heuristics and Problem Solving**. New Directions for Teaching and Learning. 2003. 53 - 58. 10.1002/tl.113.

Banco Central do Brasil. **Sistema de Pagamentos Brasileiro (SPB) - Estatísticas**. 2023. Recuperado de <https://www.bcb.gov.br/estatisticas/spbadendos>.

CINTRA, Glauber, MIYAZAWA, Flavio, WAKABAYASHI, Yoshiko, XAVIER, Eduardo. **Algorithms for two-dimensional cutting stock and strip packing problems using dynamic programming and column generation**. European Journal of Operational Research. 2008. Disponível em:
<<https://www.sciencedirect.com/science/article/abs/pii/S0377221707008831>>

DAS, Swagatam, & SUGANTHAN, Ponnuthurai. (2011). **Differential Evolution: A Survey of the State-of-the-Art**. IEEE Transactions on Evolutionary Computation, 15(1), 4-31. DOI: 10.1109/TEVC.2010.2059031.

DEB, Kalyanmoy, JAIN, Himanshu. **An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints**. IEEE Transactions on Evolutionary Computation, v. 18, n. 4, p. 577-601, 2014.

DE CARVALHO, Charliny Barros; SOLIANI, Rodrigo Duarte; DE FREITAS, César Gomes. **Logística de transporte de valores: uma revisão sistemática da literatura: Cash**. Brazilian Journal of Development, p. 55241-55258, 2022.

EL-SHERBENY, Nasser. (2010). **Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods**. Journal of King Saud University - Science, 22(3), 123-131. <https://doi.org/10.1016/j.jksus.2010.03.002>.

GÓMEZ, Luis Alberto, et al. **Aplicação da programação de geometria genética na otimização de perfis de vigas**. International Journal of Knowledge Engineering and Management, v. 3, n. 5, p. 40-53, 2014.

HAUSMAN, Warren, Lee, Hau. and Subramanian, Uma. **The Impact of Logistics Performance on Trade**. Prod Oper Manag, 22: 236-252. 2013. Recuperado de <https://doi.org/10.1111/j.1937-5956.2011.01312.x>.

HOLLAND, John. (1992). **Adaptation in Natural and Artificial Systems**. MIT Press.

LOPES, Ana Lúcia, GALVÃO, Ana Lúcia, FOGAÇA, Moacir. **Pesquisa Operacional**. 2015. Recuperado de <https://repositorio.animaeducacao.com.br/bitstreams/f362be74-a176-4495-91b0-36f6a74de72c/download>.

LOPES, Luciana de Sousa. **Uma heurística baseada em algoritmos genéticos aplicada ao problema de cobertura de conjunto**. Dissertação de Mestrado em Computação Aplicada–INPE. 1995.

MITCHELL, Melanie. **Genetic algorithms: An overview**. In: Complex. 1995. p. 31-39. Nubank. Como é feito um cartão de crédito? Blog Nubank. 2023. Recuperado de <https://blog.nubank.com.br/como-e-feito-cartao-de-credito/>.

MUTHURAMAN, Sangeetha & VENKATESAN, V.. (2017). **A Comprehensive Study on Hybrid Meta-Heuristic Approaches Used for Solving Combinatorial Optimization Problems**. 185-190. 10.1109/WCCCT.2016.53.

OLIVEIRA, Jéssica Moia de. **O PROBLEMA DE TRANSPORTE**. Campinas, SP, 2016. Disponível em: <https://www.ime.unicamp.br/~mac/db/2016-1S-140692.pdf>. Acesso em: 14 jun. 2024.

OLIVEIRA, José & Carravilla, Maria. (2009). FEUP. **Combinatorial Optimization Heuristics and Local Search**. Disponível em: <https://paginas.fe.up.pt/~mac/ensino/docs/OR/CombinatorialOptimizationHeuristicsLocalSearch.pdf>.

PEDREIRA, Patrick. **Impaciência do Consumidor em Tempos de Internet Rápida**. LinkedIn. 2016. Recuperado de <https://www.linkedin.com/pulse/impaciencia-do-consumidor-em-tempos-de-internet-rapida-pedreira/?originalSubdomain=pt>.

PIMENTEL, Renato. **Problemas de transporte**. Uberlândia, MG. 2020. Disponível em: <https://www.facom.ufu.br/~rpimentel/files/gsi027-2020-2/gsi027.transporte.pdf>. Acesso em: 14 jun. 2024.

PINTO, Lucas Thomaz Januario et al. **Um Modelo Matemático para Redução de Custos com Manufatura e Distribuição de Cartões de Crédito**. Anais, 2023.

PLOUS, S. **The psychology of judgment and decision making** New York: McGrawHill. 1993.

ROCHA, Giulia Rossatto. **Aplicação de algoritmo genético para roteirização e carregamento de veículo**. 2023.

SANTOS NETO, José Eraldo dos, et al. **Desenvolvimento de um modelo de algoritmo genético para otimização do custo na fase de transporte logístico do pequeno empreendedor**. 2023.

TALBI, El-Ghazali. (2009). **Metaheuristics: From Design to Implementation**. John Wiley & Sons. Disponível em: https://zeus.inf.ucv.cl/~bcrawford/DOCTORADO-METAHEURISTICAS/Cap1_Metaheuristics_Talbi.pdf.

TONETTO, Leandro, Kalil, Lisiane, Melo, Wilson, Schneider, Daniela, & Stein, Lilian. **O papel das heurísticas no julgamento e na tomada de decisão sob incerteza**. Estudos De Psicologia (campinas), 23(2), 181–189. 2006. <https://doi.org/10.1590/S0103-166X2006000200008>.

YANG, Xin-She. (2010). **Nature-Inspired Metaheuristic Algorithms**. Research Gate.