RBCTI
Revista Brasileira de Ciência, Tecnologia e Inovação

DOI: https://doi.org/10.18554/rbcti.v6i1.5403

Artigo

Otimização do estoque de uma fábrica de bebidas não alcoólicas em Python

Inventory Optimization of a soft drink factory in Python

Breno Prado Franchini Bizerra[1]*, Vinicius Abrão da Silva Marques[2*]

*Departamento de Engenharia Mecânica, Universidade Federal do Triângulo Mineiro, Uberaba, MG, Brasil.
[1]Orcid: 0000-0002-0078-0401 E-mail: breno.bizerra@gmail.com, [2]Orcid: 0000-0002-5499-0404 E-mail: vinicius.marques@uftm.edu.br

RESUMO: Este trabalho tem o intuito de apresentar a utilização de modelagem matemática computacional para otimização em Python do estoque de uma fábrica. O problema é construído em torno da organização da produção de bebidas. A utilização dos algoritmos *Scipy.optimize.minimize* e Algoritmo Genético (*Genetic Algorithm*) empregados neste trabalho tem como objetivo otimizar a quantidade de produtos em estoque, levando em conta os diferentes produtos fabricados, a previsão de venda baseada em histórico e a capacidade máxima de produção. Os métodos utilizados são comparados em suas respostas e tempo de processamento. Os resultados mostram que ambos os métodos têm a capacidade de encontrar respostas otimizadas para o problema. O método *Genetic Algorithm* encontra respostas menos otimizadas para cada produto, mas realiza uma melhor distribuição geral da produção, além de necessitar de maior tempo de processamento para analisar todas as possibilidades. O método *Scipy.optimize.minimize* encontra uma resposta mais otimizada para a maioria dos produtos, mas não realiza uma distribuição eficiente da produção, além de apresentar um menor tempo de processamento.

Palavras-chave: Otimização de Estoque, Previsão de vendas, Python, Scipy, Algoritmo Genético.

ABSTRACT*: The aim of the present article is to present computational mathematical modeling using to optimize a company's inventory based on Python. The study concerned soft drink production organization. Algorithms Scipy.optimize.minimize and Genetic Algorithm were herein used to optimize the quantity of products in stock by taking into consideration different manufactured goods, forecast sales based on sales history, and production maximum capacity. The adopted methods were compared to assess their responses and processing time. Based on the results, both methods are capable of giving optimal responses. The Genetic Algorithm method gives lesser optimized answers to each product, but it accounts for better general production distribution, although it needs more processing time to analyze all possibilities. The Scipy.optimize.minimize method provided more optimized answer to most products, but it was not efficient in production distribution, although it presented shorter processing time.*

Keywords*: Inventory Optimization, Sales forecast, Python, Scipy, Genetic Algorithm.*

RBCTI
Revista Brasileira de Ciência, Tecnologia e Inovação

INTRODUÇÃO

Com o notável crescimento da produção de bebidas não alcoólicas no Brasil nos últimos anos torna-se necessária a existência de maneiras mais eficientes de organização da produção e maior eficiência do cronograma de funcionamento de maquinários, aumentando a produtividade das indústrias (PAGLIARUSSI; MORABITO; SANTOS, 2016). O crescimento do número de fábricas de pequeno e médio porte levou a uma maior concorrência e crescente exigência de qualidade na categoria. Mas como estar na frente do mercado? (SEBRAE, 2021).

O emprego de ferramentas computacionais de simulação da produção, sua modelagem e criação de métodos de otimizações matemáticas, permitem que a empresa diminua o seu tempo de *setup* do maquinário e aumente a eficiência do seu cronograma de produção. Sendo assim, têm-se uma diminuição do custo envolvido no processo de fabricação e consequentemente custo final do produto.

Uma das técnicas mais utilizadas de otimização matemática computacional é o *Genetic Algorithm*. Trata-se de um processo evolucionário, ou seja, se trabalha com uma população de candidatos (possíveis respostas ao problema) em paralelo, sendo possível a busca de diferentes soluções e com alocação de membros em vários espectros de respostas. Por serem baseados em evoluções biológicas, trabalham com o método “gerar e testar” para chegar em soluções ótimas ou quase ótimas, sem grandes limitações. (CARVALHO, 2021).

Outra técnica computacional para otimizações matemáticas que se destaca por sua simplicidade de implementação é o *Scipy.optimize.minimize* o qual permite encontrar os valores de mínimos, máximos ou zeros de uma função (SCIPY-LECTURES, 2020).

Para a aplicação dos métodos citados na otimização de uma fábrica de bebidas faz-se necessário inicialmente ter o conhecimento dos processos envolvidos e da gama de produtos fabricados. Em seguida propõem-se então soluções que satisfazem de forma eficiente as demandas da empresa. (PESSANHA; ALVARENGA; ARICA, 2015).

O objetivo deste trabalho foi otimizar o estoque de uma empresa de bebidas não alcoólicas, utilizando de dados fornecidos pela empresa, como seu histórico de vendas e controles de estoque. Com a aplicação dos métodos de otimização e baseado nos dados fornecidos, foi possível implementar um algoritmo que analisa a previsão de vendas e gerencia o estoque, sem que a metodologia seja invasiva ao funcionamento das máquinas da fábrica.

Levando em conta que a variação de vendas dos produtos funciona de forma sazonal, pode-se estimar uma produção necessária baseada nas vendas de meses anteriores e assim evitar que ocorram falta de produtos nos estoques ou também um excesso de produção, ambos prejudiciais ao orçamento da empresa.

As vantagens da aplicação do método utilizado neste trabalho são o baixo custo da implementação proposta, baseada em linguagem Python, um *software* livre. Além disso não houve necessidade de visitas presenciais à empresa, por utilizar apenas os bancos de dados históricos da empresa.

Como resultados deste trabalho têm-se a diminuição de custo gerada indiretamente por um melhor controle de estoques, melhorando sua organização de produção, diminuindo horas de trabalho e de *setup* de maquinário.

RBCTI
Revista Brasileira de Ciência, Tecnologia e Inovação

PROCEDIMENTOS METODOLÓGICOS

Em busca de uma solução mais próxima de um ótimo nominal para o problema proposto neste trabalho, todo o funcionamento da linha de produção de uma indústria de bebidas não alcoólicas reais, localizada no estado de São Paulo, foi analisado. Foi feito a modelagem matemática do controle de estoque da empresa baseado em históricos de vendas. Outros parâmetros foram também mapeados como a capacidade máxima de produção, a qual está ligada à quantidade de garrafas que a máquina responsável por moldar as garrafas pets (sopradora) conseguem produzir. Foi verificado este processo como sendo o mais crítico da fábrica, sendo então utilizado para encontrar a função objetivo e suas restrições, descritas a seguir, as quais são utilizadas pelos métodos de otimização. Buscando o mínimo global de uma função objetivo limitado por restrições é possível encontrar a melhor solução possível e real para o problema.

Restrições

Com os dados de limite da produção, estoques limiares e vendas previstas foi possível encontrar equações matemáticas que representam as restrições reais da fábrica. No equacionamento das restrições foram criadas duas variáveis dependentes, denominadas $rest1$ e $rest2$, definidas de maneira a ser implementadas por cada um dos dois métodos estudados.

Para cada rotina de código, foram criadas listas de quantidades de produto. Essas listas possuem os valores previstos de venda (Vp), que são baseados nas vendas do ano anterior. Em seguida são calculados os demais parâmetros de entrada dos algoritmos dados pelo estoque inicial (Ei – equação 1), estoque desejado (Ed – equação 2) e limiar de estoque (L – equação 3), respectivamente definidos proporcionalmente à Vp e conforme esperado pela equipe gestora da empresa.

$$Ei = Vp \times 0.6 \quad (1)$$
$$Ed = Vp \times 0.7 \quad (2)$$
$$L = Vp \times 0.1 \quad (3)$$

Nas restrições do algoritmo de minimização *Scipy.optimize.minimize* foram utilizadas funções de desigualdade, as quais devem ser não negativas, ou seja, conforme definido pela biblioteca do método, os valores das variáveis de restrição devem ser maiores ou iguais a zero. A partir dos dados de previsão de venda para cada produto, contidos no vetor Vp, é possível calcular a quantidade "x" de cada produto "j" que deverá ser produzida para suprir a demanda.

$$Ef[j] = Ei[j] - Vp[j] + x[j] \quad (4)$$
$$rest1[j] = Ef[j] - L[j] \quad (5)$$
$$rest2 = P\_max - np.sum(x) \quad (6)$$

No qual "Ef", equação 4, é o estoque final do ciclo de produção, dado pela diferença do estoque inicial mais a produção menos a previsão de venda; P_max é a produção máxima real da fábrica; e $np.sum(x)$ (comando da biblioteca *Numpy*) é a soma da produção de todos os produtos. Na primeira restrição, equação 5, a diferença do estoque final e o

RBCTI
Revista Brasileira de Ciência, Tecnologia e Inovação

limiar do estoque deve ser sempre maior ou igual a zero. Na segunda restrição, equação 6, a soma de produção de todos os produtos deve ser menor ou igual ao limite de produção da fábrica.

Para o método de *Genetic Algorithm*, as restrições se baseiam nas mesmas informações, porém enquanto a primeira restrição é ainda obtida pela equação 5, a segunda restrição é obtida pela equação 7, de maneira a cumprir com as exigências de funcionamento do algoritmo, neste caso:

$$rest2 = np.sum(x) - P\_max \quad (7)$$

Os algoritmos buscam otimizar a diferença entre o estoque final e o desejado variando a quantidade de produtos produzidos, sendo que a resposta deve ser buscada dentro de um intervalo delimitado pelas restrições de desigualdade, até convergir para o melhor resultado possível. Para que as restrições criadas façam com que o sistema convirja para um resultado plausível com o real, foi criado um sistema de penalidades, no qual as respostas encontradas que estejam fora dos limites das restrições são igualadas a um valor elevado. Neste caso, 10^{10} foi a penalidade escolhida para que o usuário saiba que a resposta recebida foi penalizada, ou seja, o código descarta a solução e gera uma nova interação para buscar uma solução otimizada.

Tendo definido o sistema de equações de restrições para os dois algoritmos, pode-se partir para a modelagem da função objetivo.

Função objetivo

A função objetivo é calculada a partir da mesma base de dados utilizada nas restrições, ou seja, utiliza da formulação de Estoque final (equação 4) e Estoque desejável (equação 2) para otimizar a diferença entre os dois parâmetros.

$$Var = np.sum(np.abs(Ed[j] - Ef[j])) \quad (8)$$

O problema se resume a uma única função objetivo denominada Var, conforme equação 8, a qual é obtida pela soma, para todos os produtos fabricados, dos módulos das diferenças entre o estoque desejado e o estoque final. A mesma formulação é utilizada para os dois algoritmos. Portanto o algoritmo irá buscar a produção $x[j]$ que minimize o valor de Var, respeitando as limitações implementadas nas restrições.

Caso ocorra de o sistema não convergir, é possível alterar os *inputs* de número de iterações e tamanho da população para que haja mais possibilidades de o algoritmo encontrar uma resposta que convirja. Como, por exemplo, quando o número de iterações é muito baixo, não há a possibilidade de o código convergir para um resultado otimizado.

RESULTADOS E DISCUSSÃO

Resultados Computacionais

Nesta seção do trabalho serão apresentados os resultados obtidos na implementação dos dois métodos de otimização (*Genetic algorithm* e *Scipy.optimize.minimize*).

Foram utilizados respectivamente as seguintes bibliotecas do Python:

RBCTI
Revista Brasileira de Ciência, Tecnologia e Inovação

1- geneticalgorithm [1];
2- scipy.optimize com o método SLSQP [2].

Os valores dos parâmetros de *setup* foram escolhidos a partir de uma análise de sensibilidade para cada um dos algoritmos. Visto que o objetivo deste trabalho não é discutir a influência dos parâmetros, e sim os resultados do estudo de caso abordado, são apresentados os parâmetros e resultados do caso, dentre os que foram simulados, que se mostrou o mais otimizado. Mais otimizado é considerado o caso que possui menor valor da função objetivo (equação 8). Isso quer dizer que a escolha dos parâmetros foi independente do tempo de processamento. Os resultados do tempo de processamento são apresentados apenas com o objetivo de comparar o desempenho computacional dos métodos, tendo como referência as simulações em um computador Intel Core i7 com 8 Gb de memória RAM.

Para o Algoritmo Genético (*Genetic Algorithm*), os parâmetros de inicialização do código foram determinados como demonstram as equações de 9 a 13.

$$n = 22 \quad (9)$$
$$Ea = np.zeros(n) \quad (10)$$
$$Var = np.zeros(n) \quad (11)$$
$$rest1 = np.zeros(n) \quad (12)$$
$$rest2 = np.zeros(n) \quad (13)$$

Os parâmetros internos escolhidos no Algoritmo genético para entregar a resposta mais otimizada dentro da capacidade de processamento do computador utilizado foram escolhidos da forma como está apresentado nas equações de 14 a 21.

$$'max\_num\_iteration': 800 \quad (14)$$
$$'population\_size': 800 \quad (15)$$
$$'mutation\_probability': 0.2 \quad (16)$$
$$'elit\_ratio': 0.01 \quad (17)$$
$$'crossover\_probability': 0.3 \quad (18)$$
$$'parents\_portion': 0.3 \quad (19)$$
$$crossover\_type': 'uniform' \quad (20)$$
$$'max\_iteration\_without\_improv': None \quad (21)$$

Para o algoritmo *Scipy.optimize.minimize*, foram utilizados os mesmos parâmetros iniciais determinados para o Algoritmo Genético e os parâmetros internos foram definidos dentro das opções dadas pelo método “SLSQP” como é apresentado nas equações 22 e 23.

$$'type': 'ineq', 'fun': restricao1 \quad (22)$$
$$'type': 'ineq', 'fun': restricao2 \quad (23)$$

[1] Disponível em: https://pypi.org/project/geneticalgorithm/
[2] Disponível em: https://docs.scipy.org/doc/scipy/reference/optimize.minimize-slsqp.html

RBCTI
Revista Brasileira de Ciência, Tecnologia e Inovação

As informações utilizadas foram coletadas no banco de dados da empresa e informações técnicas dos responsáveis da produção. Este estudo leva em conta todas as limitações reais da fábrica analisada, para que não haja divergências no resultado obtido neste trabalho e as possibilidades de implementação na fábrica. Para o desenvolvimento deste trabalho considerou-se as vendas do mês de outubro de 2020 como base de dados inicial e a produção total de 430.000 garrafas produzidas em funcionamento normal das máquinas.

A **Tabela 1** fornece a quantidade de garrafas vendidas de cada produto produzido na empresa no período do mês de outubro. Com estes dados é possível obter os valores de estoque inicial, estoque desejado e limiar de estoque para aplicar nas formulações do código.

Tabela 1. Quantidade de garrafas vendidas de cada produto no mês de outubro.

	PRODUTO	QUANTIDADE DE GARRAFAS
1	Guaraná 2L	37674
2	Gipps guaraná 2L	16872
3	Laranja 2L	18702
4	Limão 2L	9186
5	Citrus 2L	12474
6	Cola 2L	37980
7	Água c/ gás 2L	9180
8	Água c/ gás limão 1L	4716
9	Água c/ gás 510ml	42120
10	Água c/ gás limão 510ml	97380
11	Água c/ gás maçã 510ml	30876
12	Guaraná 250ml	21732
13	Laranja 250ml	4620
14	Maçã 2L	38586
15	Uva 2L	11406
16	Uva 250ml	4884
17	Cola 250ml	6756
18	Cola 1L	2592
19	Guaraná 1L	6660
20	Maçã 1L	11784
21	Maça 250ml	17652
22	Energético k10	1800

A partir dos dados da **Tabela 1** é possível encontrar também a produção ideal, baseada no estoque inicial e a previsão de venda, que seriam de 490.195 garrafas. Esta produção ideal indica qual deveria ser a produção de cada item de maneira a resultar em uma quantidade em estoque igual à desejada. No entanto, com a produção máxima possível pela fábrica de 430.000 garrafas/mês, é necessário que haja uma otimização da produção para encaixar a melhor quantidade de produção de cada produto e assim minimizar a diferença entre estoque desejado e estoque final.

RBCTI
Revista Brasileira de Ciência, Tecnologia e Inovação

Para o algoritmo *Scipy.optimize.minimize* e para o *Genetic Algorithm*, foram encontrados os resultados de produção necessária para otimizar o estoque apresentados na **Tabela 2**.

Pode-se observar na **Tabela 2** que as produções necessárias encontradas pelos dois algoritmos buscam, de maneiras diferentes, encontrar o estoque final mais próximo do estoque desejado. Apesar do algoritmo *Scipy.optimize.minimize* ter um tempo de processamento de cerca de 1 segundo, observa-se que o *Genetic Algorithm*, no qual o tempo de processamento é de cerca de 1 minuto e meio, tem maior eficácia, pois encontra não apenas um resultado mais otimizado, e sim um resultado com melhor aplicabilidade real, se adaptando à produção necessária de cada produto.

Na **Tabela 3** é apresentado um comparativo dos valores de estoque desejado com os valores de estoques finais resultantes da aplicação de ambos os métodos.

Tabela 2. Produção otimizada de cada produto.

	PRODUTO	SCIPY MINIMIZE	GA
1	Guaraná 2L	41441	29334
2	Gipps guaraná 2L	18559	15725
3	Laranja 2L	20572	17562
4	Limão 2L	10104	7238
5	Citrus 2L	13721	12816
6	Cola 2L	41777	35068
7	Água c/ gás 2L	10097	8121
8	Água c/ gás limão 1L	5187	3857
9	Água c/ gás 510ml	46331	36967
10	Água c/ gás limão 510ml	46922	106227
11	Água c/ gás maçã 510ml	33963	31629
12	Guaraná 250ml	23905	21524
13	Laranja 250ml	5081	4302
14	Maçã 2L	42444	39519
15	Uva 2L	12546	12444
16	Uva 250ml	5372	3176
17	Cola 250ml	7431	5743
18	Cola 1L	2851	1802
19	Guaraná 1L	7325	4856
20	Maçã 1L	12962	11401
21	Maça 250ml	19417	19403
22	Energético k10	1979	1264
	Total	429987	429978

RBCTI
Revista Brasileira de Ciência, Tecnologia e Inovação

Tabela 3. Comparação dos Estoques finais otimizados com o Estoque desejado.

	PRODUTO OUTUBRO	ESTOQUE DESEJADO	ESTOQUE FINAL SCIPY.MINIMIZE	ESTOQUE FINAL GENETIC ALGORITHM
1	Guaraná 2L	26371	26371	14265
2	Gipps guaraná 2L	11810	11810	8977
3	Laranja 2L	13091	13091	10081
4	Limão 2L	6430	6430	3564
5	Citrus 2L	8731	8731	7828
6	Cola 2L	26586	26585	19876
7	Água c/ gás 2L	6426	6425	4449
8	Água c/ gás limão 1L	3301	3301	1970
9	Água c/ gás 510ml	29484	29483	20119
10	Água c/ gás limão 510ml	68166	7970	67275
11	Água c/ gás maçã 510ml	21613	21613	19279
12	Guaraná 250ml	15212	15212	12831
13	Laranja 250ml	3234	3233	2454
14	Maçã 2L	27010	27010	24085
15	Uva 2L	7984	7984	7882
16	Uva 250ml	3418	3418	1222
17	Cola 250ml	4729	4729	3040
18	Cola 1L	1814	1814	765
19	Guaraná 1L	4662	4661	2192
20	Maçã 1L	8248	8248	6687
21	Maça 250ml	12356	12356	12342
22	Energético k10	1260	1259	544
	Total	311936	251734	215727

Na **Tabela 3** é verificado que o algoritmo *Scipy.optimize.minimize* busca a melhor otimização de produção necessária para cada produto, mesmo que acabe faltando espaço de produção para um dos produtos. Já para o *Genetic Algorithm*, pode-se observar uma melhor interpretação da necessidade da fábrica, fazendo com que a produção possa ser menor em cada produto, mas sempre suprindo uma prioridade de produção, ou seja, buscando um resultado mais plausível com a realidade.

Verifica-se na Tabela 2 que a produção total de acordo com cada algoritmo está próxima ao limite da fábrica de 430 mil garrafas, portanto não seria possível aumentar a produção. Porém o objetivo é minimizar a diferença entre o estoque desejado e o estoque final, conforme Tabela 3. Ao analisar a Tabela 3 observa-se que o valor total de acordo com cada algoritmo está abaixo do desejado, porém não é possível aumentar a produção conforme visto na Tabela 2. Caso a produtividade máxima da fábrica fosse maior essa diferença do total dos estoques finais para o desejado seria menor, sendo que o algoritmo busca otimizar exatamente para que a diferença seja a mínima possível, levando em conta as restrições. A quantidade de produtos fabricados somado ao estoque inicial atende ainda a previsão de venda da Tabela 1, caso contrário o estoque final seria negativo, o que não foi observado. A limitação de produção máxima afeta apenas o estoque final, não se tratando, portanto, de uma perda de vendas, e sim um controle de estoque levando em conta não apenas a quantidade de garrafas produzidas, mas também o estoque inicial e a

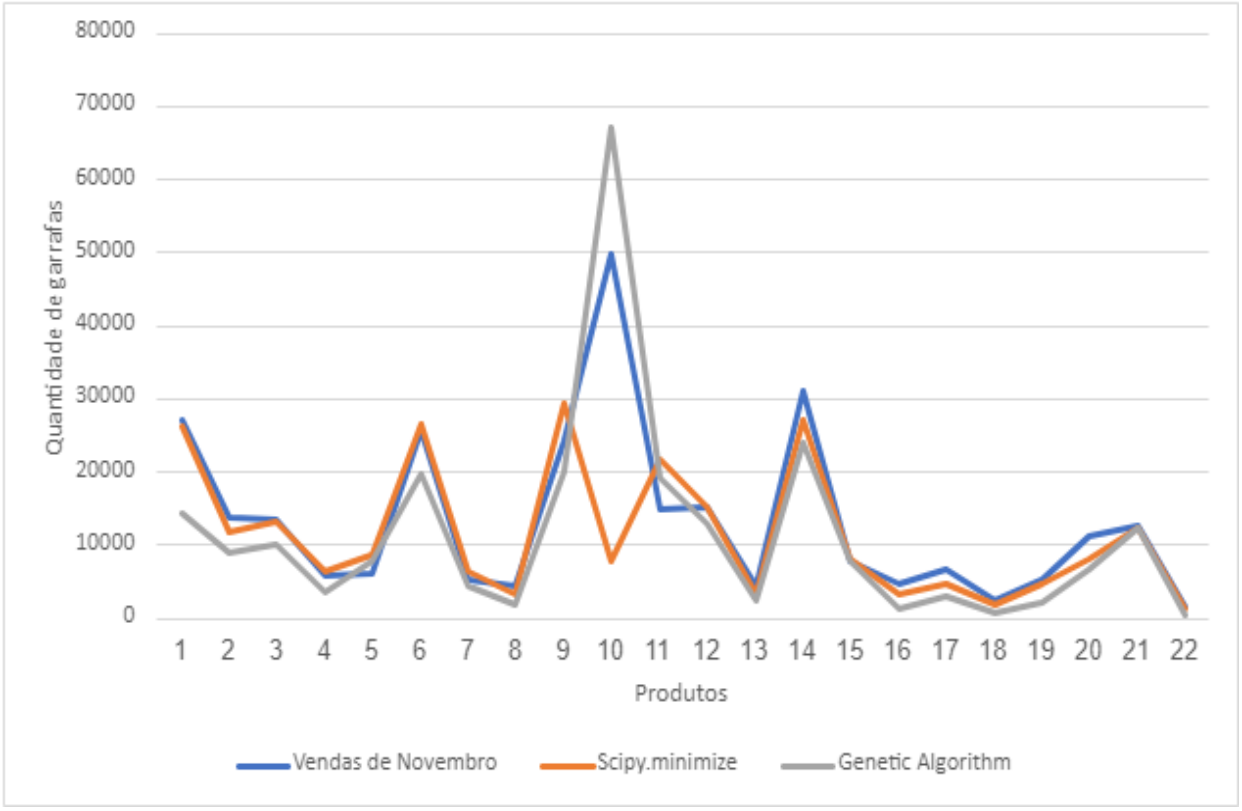
RBCTI
Revista Brasileira de Ciência, Tecnologia e Inovação

previsão de vendas, obtidos pelo histórico de vendas de outubro apresentado na Tabela 1. Por fim a restrição de estoque final acima do limiar também foi atendida por ambos os algoritmos.

Comparativo entre resultados otimizados de estoque final em outubro e vendas reais de novembro

Após a obtenção dos resultados otimizados pode-se comparar os dados de estoque final do mês de outubro previsto por ambos os métodos, ou seja, o estoque inicial do mês de novembro, com os valores reais de vendas realizadas pela fábrica no mês de novembro de 2020, conforme **Figura 1**. Desta forma é possível compreender de maneira ainda mais aprofundada o comportamento de cada algoritmo e assim definir se os algoritmos seriam confiáveis em gerir a produção de maneira compatível com as vendas do mês seguinte. Analisando a **Figura 1** pode-se verificar que o *Genetic Algorithm* resulta em um estoque mais seguro, isto é, garantindo a demanda de novembro mesmo em situações onde as vendas de um certo produto saiam da média. O mesmo não é observado na curva do *Scipy.optimize.minimize*, onde o gráfico é extremamente fiel às vendas de Novembro em quase todos os produtos, porém não consegue se adaptar e atender as situações onde as vendas têm um comportamento diferente do esperado, como o produto 10.

Figura 1. Gráfico comparativo entre estoque final otimizado de outubro e vendas de novembro.



Conclui-se que o *Genetic Algorithm* apresenta um melhor resultado pois atende melhor em casos que não se tem uma uniformidade nas vendas entre os produtos.

RBCTI
Revista Brasileira de Ciência, Tecnologia e Inovação

Novamente destaca-se que a Figura 1 se trata de uma análise de estoque inicial com vendas previstas, ou seja, não se trata de dizer que o estoque não atenderá as vendas de novembro. O que é observado na Figura 1 é que o estoque final de outubro otimizado pelo *Genetic Algorithm* por si só quase supriria as vendas de novembro, sendo considerado um estoque seguro. O algoritmo pode então ser novamente empregado para prever a produção do mês de novembro que suprirá a diferença do estoque inicial, as vendas previstas e ainda minimizar a diferença para que no final de novembro o estoque final seja o mais próximo possível ao desejado, dentro das restrições. O processo se repete para os meses consecutivos.

Destaca-se que a diferença no comportamento do produto 10 surge ao comparar o resultado do *Scipy.optimize.minimize* com as vendas futuras, ou seja, no momento da otimização os dados das vendas de novembro não são conhecidos. Portanto, não se trata de uma divergência do método e sim de duas interpretações diferentes de possíveis distribuições da produção. Os resultados do *Genetic Algorithm* resultam em um comportamento com menor diferença para o produto 10 pois no mês de Outubro a produção otimizada foi maior que a do *Scipy.optimize.minimize*, sendo, portanto, ambas soluções válidas e ambas atendem as restrições impostas.

CONCLUSÕES

Neste trabalho foram utilizados dois métodos de otimização computacionais para a resolução de um problema de controle de estoque de uma empresa de bebidas. Ambos os algoritmos implementados obtiveram resultados otimizados, porém com aplicabilidades diferentes.

Visto que não é de conhecimento dos autores nenhum estudo de caso semelhante apresentado em trabalhos científicos presentes na literatura, foram utilizados neste trabalho os dados reais de uma fábrica e assim analisados os resultados obtidos pelos modelos de otimização propostos.

Comparativamente, o algoritmo *Scipy.optimize.minimize* tem seu tempo de processamento dezenas de vezes mais rápido. Porém sua desvantagem está no fato de não compreender a distribuição de cada produto na produção geral, podendo fazer com que um produto de alta demanda tenha sua produção debilitada.

O *Genetic Algorithm* por sua vez, consegue encontrar uma produção suficientemente otimizada e dentro dos parâmetros de demanda de produtos, pois analisa diversas possíveis respostas e busca a que se encaixa melhor com a distribuição de cada produto nas vendas gerais. Porém, para que haja eficiência no código, é necessário que o número de iterações e o tamanho da população do código sejam grandes, ocasionando em um maior tempo de processamento.

Conclui-se que o *Genetic Algorithm* apresenta melhores resultados, pois atende melhor em casos que não se tem uma uniformidade nas vendas entre os produtos, garantindo um estoque final mais seguro para atender o mês seguinte.

Com as soluções obtidas é possível otimizar diferentes necessidades do mercado, aumentando a rentabilidade do negócio e sem que haja grandes custos operacionais.

A sugestão dos autores para trabalhos futuros é a aplicação da metodologia proposta ao longo de um ano de produção da fábrica de maneira a identificar possíveis influências de algum parâmetro não mapeado e assim poder aprimorar o modelo utilizado.

RBCTI
Revista Brasileira de Ciência, Tecnologia e Inovação

REFERÊNCIAS

CARVALHO, A. P. L. F. **Algoritmos genéticos**. Disponível em: https://sites.icmc.usp.br/andre/research/genetic/. Acesso em: 10 jan. 2021.

MALAQUIAS, N G. L. **Uso dos algoritmos genéticos para a otimização de rotas de distribuição**. 2006. 113p. Tese (Mestrado em Engenharia Elétrica) - Universidade Federal de Uberlândia, Faculdade de Engenharia Elétrica, Uberlândia, MG, 2006.

PAGLIARUSSI, M. S.; MORABITO, R; SANTOS, M. O. Otimização da programação da produção de bebidas à base de frutas por meio de modelos de programação inteira mista. **Scientific Electronic Library Online – Scielo,** p. 1, dezembro 2016. DOI: https://doi.org/10.1590/0104-530X2288-15. Disponível em: https://www.scielo.br/j/gp/a/XM8DcbSWFGDHCTPfHWJWrCB/?lang=pt. Acesso em: 09 jan. 2021.

PESSANHA, L. P. M.; ALVARENGA, R. L.; ARICA, G. G. M. Modelagem e resolução do problema integrado de dimensionamento e sequenciamento da produção: Caso de uma pequena empresa de produtos de limpeza. *In*: **XXXV Encontro nacional de engenharia de produção**. Fortaleza, CE, 2015.

SCIPY-LECTURES. **Mathematical optimization: finding minima of functions**. Disponível em: https://scipy-lectures.org/advanced/mathematical_optimization/. Acesso em: 23 dez. 2020.

SEBRAE. **Micro e pequenas empresas geram 27% do PIB do Brasil.** Disponível em: https://www.sebrae.com.br/sites/PortalSebrae/ufs/mt/noticias/micro-e-pequenas-empresas-geram-27-do-pib-do-brasil,ad0fc70646467410VgnVCM2000003c74010aRCRD. Acesso em: 09 jan. 2021.

Recebido em: 12/04/2021
Aprovado em: 04/07/2021