



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO
CURSO DE GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

LEANDRO HENRIQUE GOMES DA SILVA

**APLICAÇÃO DO ALGORITMO GENÉTICO E SIMULAÇÃO DE EVENTOS
DISCRETOS EM UMA INDÚSTRIA DE LATAS**

Recife

2024

LEANDRO HENRIQUE GOMES DA SILVA

**APLICAÇÃO DO ALGORITMO GENÉTICO E SIMULAÇÃO DE EVENTOS
DISCRETOS EM UMA INDÚSTRIA DE LATAS**

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Produção da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do grau em Engenharia de Produção.

Orientador: Dr. Márcio José das Chagas Moura

Recife

2024

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Silva, Leandro Henrique Gomes da.

Aplicação do algoritmo genético e simulação de eventos discretos em
uma indústria de latas. / Leandro Henrique Gomes da Silva. - Recife, 2024.
91p. : il., tab.

Orientador(a): Márcio José das Chagas Moura

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal
de Pernambuco, Centro de Tecnologia e Geociências, Engenharia de
Produção - Bacharelado, 2024.

Inclui referências, apêndices.

1. Algoritmo genético. 2. Modelagem de falhas. 3. Indústria de latas. 4.
Otimização combinatória. 5. Simulação de eventos discretos. I. Moura, Márcio
José das Chagas . (Orientação). II. Título.

620 CDD (22.ed.)

Leandro Henrique Gomes da Silva

APLICAÇÃO DO ALGORITMO GENÉTICO E SIMULAÇÃO DE EVENTOS DISCRETOS EM UMA INDÚSTRIA DE LATAS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Produção da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do grau em Engenharia de Produção.

Aprovado em: 19/03/2024

BANCA EXAMINADORA

Prof. Dr. Márcio José das Chagas Moura (Orientador)
Universidade Federal de Pernambuco

Prof. Dra. Isis Didier Lins (Examinador Interno)
Universidade Federal de Pernambuco

Prof. Dr. Marcelo Hazin Alencar (Examinador Interno)
Universidade Federal de Pernambuco

AGRADECIMENTOS

Gostaria de expressar minha gratidão a todas as pessoas que contribuíram de diversas maneiras para a realização deste trabalho acadêmico.

Primeiramente, gostaria de agradecer ao meu orientador Márcio Chagas, pela orientação dedicada, paciência e valiosos insights fornecidos ao longo deste processo. Suas aulas foram fundamentais para a tomada de decisão sobre a minha área de atuação profissional.

Expresso minha gratidão aos meus familiares, especialmente a minha mãe Conceição e avó Angelina, que fizeram o possível e impossível para que eu chegasse aonde estou. Aos meus amigos Vinícius, Ubiratan e João por estarem presentes desde minha entrada até a conclusão do curso. E a minha namorada Rebeca por seu apoio diário.

Gostaria de estender meus agradecimentos a Lucas, William, Hugo, Rafael e Cláudio que contribuíram para a realização deste trabalho, seja por meio de colaborações, sugestões ou incentivos.

Por fim, gostaria de agradecer ao apoio do Instituto de Inovação Tecnológica, por disponibilizar computadores para obter os resultados deste trabalho e a empresa por confiar e disponibilizar dados reais para estudo.

A todos vocês, meu mais sincero obrigado.

Leandro Henrique Gomes da Silva

RESUMO

O uso de softwares de simulação vem sendo cada vez mais comum para o apoio a tomada de decisão, porém em algumas situações a busca por soluções de problemas complexos pode se tornar inviável por questão de tempo para obter-se uma solução, um exemplo disso é a busca por um conjunto de posicionamentos ideais para sensores ao longo da linha de produção. Este estudo aborda a otimização do fluxo de latas do modelo de simulação da indústria, resolvendo o problema de posicionamento dos sensores localizados em esteiras entre as máquinas, responsáveis por tomar decisão de bloquear ou desbloquear a entrada de uma máquina de acordo com a lotação da esteira atual, para impedir que acúmulos de latas na esteiras ocorram, que por sua vez implica em vários outros problemas. Buscando formas de tornar a solução destes tipos de problemas viáveis experimentou-se a combinação da otimização combinatória e simulação de produção para encontrar o posicionamento ideal dos sensores, além de preencher lacunas na modelagem de falhas abertas no modelo de simulação utilizado. Neste trabalho, foi escolhido o algoritmo genético por ser uma ferramenta embutida do software de simulação utilizado pela empresa. O estudo envolveu a análise das restrições do posicionamento dos sensores e a correta utilização de hiper parâmetros para a aplicação do algoritmo genético. Também foi adicionada a modelagem de tempo médio entre falhas e tempo médio de reparos para impor condições mais realistas ao modelo de simulação, juntamente com a realização de experimentos comparativos para avaliar os métodos de configuração sugeridos pela literatura. Como resultado, verificou-se que a aplicação do algoritmo genético em conjunto com a simulação de eventos discretos pode proporcionar melhorias significativas no processo de produção de latas, pois consegue encontrar uma boa solução em tempo hábil que, de forma robusta, comprova o aumento da capacidade produtiva da indústria em, no mínimo, 0,51%, números significantes para a empresa, além de proporcionar um conjunto de soluções simples e diretas para o decisor. Os resultados sugerem uma preferência maior pelo uso da configuração Dynamic ILM/DHC do algoritmo genético devido a sua capacidade em diminuir o espaço de busca e encontrar soluções melhores e menos tempo.

Palavras-chave: Algoritmo genético, Modelagem de falhas, Indústria de latas, Otimização combinatória, Simulação de eventos discretos.

ABSTRACT

The use of simulation software is becoming increasingly common to support decision-making. However, in some situations, the pursuit of solutions to complex problems can become impractical due to time constraints in obtaining a solution. An example of this is the search for an ideal set of sensor placements along the production line. This study addresses the optimization of the can flow within the simulation model of the industry, solving the problem of positioning sensors located on conveyors between machines. These sensors are responsible for making decisions to block or unblock the entry of a machine based on the current conveyor belt occupancy, to prevent build-ups on the conveyors, which in turn leads to various other issues. In an attempt to make the solution to these types of problems feasible, the combination of combinatorial optimization and production simulation was experimented with to find the optimal sensor placement. Additionally, gaps in fault modeling in the simulation model were addressed. For this work, the genetic algorithm was chosen as the tool because it is embedded in the simulation software used by the company, facilitating its implementation. The study involved analyzing sensor placement constraints and correctly utilizing hyperparameters for the genetic algorithm application. Furthermore, the modeling of average time between failures and average repair time was added to impose more realistic conditions on the simulation model. Comparative experiments were conducted to evaluate the configuration methods suggested in the literature. As a result, it was found that the application of the genetic algorithm combined with discrete event simulation can provide significant improvements in can production processes. It manages to find a good solution in a timely manner, robustly proving the increase in the industry's production capacity by at least 0.51%, which is significant for the company. Moreover, it provides a set of simple and direct solutions for decision-makers. The results suggest a preference for the Dynamic ILM/DHC configuration of the genetic algorithm due to its ability to reduce the search space and find better solutions in less time.

Keywords: Genetic algorithm, Fault modeling, Can industry, Combinatorial optimization, Discrete event simulation.

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1 – Fluxograma de otimização em simulação | 19 |
| Figura 2 – Métodos de buscas estocásticos | 20 |
| Figura 3 – Possíveis bicos de pássaros | 21 |
| Figura 4 – Listagem de variantes do GA | 22 |
| Figura 5 – Fluxograma do algoritmo genético | 23 |
| Figura 6 – Representação de seleção do GA | 24 |
| Figura 7 – Representação da recombinação do GA | 25 |
| Figura 8 – Representação da mutação do GA | 26 |
| Figura 9 – Representação do Fluxo de atividades da lata..... | 32 |
| Figura 10 – Representação da linha de produção | 33 |
| Figura 11 – Máquina CUP..... | 34 |
| Figura 12 – Máquina BM | 35 |
| Figura 13 – Máquina WSH..... | 36 |
| Figura 14 – Máquina PRT | 37 |
| Figura 15 – Máquina forno da PRT..... | 38 |
| Figura 16 – Máquina IS | 39 |
| Figura 17 – Máquina IBO..... | 40 |
| Figura 18 – Máquina NCK..... | 41 |
| Figura 19 – Máquina LT..... | 42 |
| Figura 20 – Máquina PLT | 43 |
| Figura 21 – Esteiras com latas..... | 44 |
| Figura 23 – Fluxograma de falhas sobre a decisão de sensores | 52 |
| Figura 24 – Demonstração da modelagem de falhas atual | 53 |
| Figura 25 – Distribuições suportadas no Plant simulation | 57 |
| Figura 26 – Alocação de cluster as distribuições (Tempo de reparo) | 60 |
| Figura 27 – Alocação de cluster as distribuições para tempo entre falhas | 62 |

LISTA DE TABELAS

| | |
|---|----|
| Tabela 1 – Configurações de sensores da esteira 1 | 45 |
| Tabela 2 – Configurações de sensores da esteira 2 | 46 |
| Tabela 3 – Configurações de sensores da esteira 3 | 46 |
| Tabela 4 – Configurações de sensores da esteira 4 | 46 |
| Tabela 5 – Configurações de sensores da esteira 5 | 46 |
| Tabela 6 – Configurações de sensores da esteira 6 | 47 |
| Tabela 7 – Configurações de sensores da esteira 7 | 47 |
| Tabela 8 – Configurações de sensores da esteira 8 | 47 |
| Tabela 9 – Configurações de sensores da esteira 9 | 48 |
| Tabela 10 – Configurações de sensores da esteira 10 | 48 |
| Tabela 11 – Configurações de sensores da esteira 11 | 48 |
| Tabela 12 – Configurações de sensores da esteira 12 | 48 |
| Tabela 13 – Configurações de sensores da esteira 13 | 49 |
| Tabela 14 – Configurações de sensores da esteira 14 | 49 |
| Tabela 15 – Configurações de sensores da esteira 15 | 49 |
| Tabela 16 – Configurações de sensores da esteira 16 | 49 |
| Tabela 17 – Configurações de sensores da esteira 17 | 50 |
| Tabela 18 – Configurações de sensores da esteira 18 | 50 |
| Tabela 19 – Clusterização de falhas | 55 |
| Tabela 20 – Quantidade de posições possíveis de cada sensor | 64 |
| Tabela 21 – Posição de sensores resultante 1 | 66 |
| Tabela 22 – Posição de sensores resultante 2 | 70 |
| Tabela 23 – Posição de sensores resultante 3 | 74 |
| Tabela 24 – Posição de sensores resultante 4 | 78 |

SUMÁRIO

| | |
|--|-----------|
| 1. INTRODUÇÃO | 11 |
| 1.1. JUSTIFICATIVA E RELEVÂNCIA | 11 |
| 1.2. OBJETIVO | 12 |
| 1.3. CONTEXTO E DESCRIÇÃO DO PROBLEMA | 13 |
| 1.3.1. Contexto do problema | 13 |
| 1.3.2. Descrição do problema | 14 |
| 2. FUNDAMENTAÇÃO TEÓRICA | 17 |
| 2.1. SIMULAÇÃO | 17 |
| 2.1.1. Sistema | 17 |
| 2.1.2. Modelo | 17 |
| 2.1.3. Ambiente | 18 |
| 2.2. SIMULAÇÃO DE EVENTO DISCRETOS | 18 |
| 2.3. SIMULAÇÃO COM OTIMIZAÇÃO COMBINATÓRIA | 19 |
| 2.4. ALGORITMO GENÉTICO | 21 |
| 2.4.1. Seleção | 23 |
| 2.4.2. Recombinação | 24 |
| 2.4.3. Mutação | 25 |
| 2.4.4. Parâmetros do algoritmo genético | 26 |
| 2.5. MODELAGEM DE FALHAS | 27 |
| 3. REVISÃO DE LITERATURA | 30 |
| 4. ESTUDO DE CASO | 32 |
| 4.1. APRESENTAÇÃO DO MODELO | 32 |
| 4.2. RESTRIÇÕES DO PROBLEMA | 44 |
| 4.3. MODELAGEM DE FALHAS | 51 |
| 4.3.1. Independência entre falhas | 53 |
| 4.3.2. Agrupamento de falhas | 54 |
| 4.3.3. Restrições do software | 57 |
| 4.3.4. Tempo de reparo | 58 |
| 4.3.5. Tempo entre falhas | 60 |
| 4.4. CONFIGURAÇÃO DE HIPERPARÂMETROS | 62 |
| 4.4.1. Dynamic ILM/DHC | 63 |
| 4.4.2. Dynamic DHM/ILC | 63 |
| 4.5. ANÁLISE DE RESULTADOS | 63 |
| 4.5.1. Resultado sem falhas modeladas | 66 |
| 4.5.2. Resultados com falhas modeladas | 74 |
| 5. CONCLUSÃO | 83 |
| 5.1. LIMITAÇÕES | 83 |
| REFERÊNCIAS | 84 |
| APÊNDICE A | 89 |

1. INTRODUÇÃO

Todas as indústrias enfrentam o desafio de se manterem competitivas em seus respectivos mercados, e isso exige constantes esforços para aumentar a eficiência operacional (CRUZ, 2011). Mesmo para multinacionais, que já operam em alto nível competitivo, encontrar maneiras de aumentar ainda mais a eficiência pode ser complicado. Uma abordagem adotada por indústrias mais avançadas é o uso de modelos de simulação para analisar e identificar oportunidades de melhoria que não são facilmente perceptíveis no chão de fábrica (ALLAN SARMENTO VIEIRA; WILSON FADLO CURI, 2016). Geralmente, quanto mais complexa for a realidade da empresa, mais necessário será a simulação nas buscas por essas melhorias.

Nesse contexto, a aplicação de algoritmos de otimização combinatória surge como uma ferramenta valiosa para resolver problemas complexos enfrentados pela indústria (REGINA MITSUE AZUMA, 2021). Ao utilizar algoritmos genéticos adequadamente, é possível explorar diferentes soluções e encontrar aquelas que não apenas aumentem a produção da empresa, mas também contribuam para o alcance dos seus objetivos.

No contexto atual, é comum saber que o uso de ferramentas avançadas de simulação desempenha um papel crucial na tomada de decisões estratégicas. Este trabalho se propõe a explorar o uso do software Tecnomatix Plant Simulation em uma indústria que já o utiliza como ferramenta de apoio à decisão há mais de sete anos e explorar também sua capacidade de otimizar um modelo de simulação através do *Genetic Algorithm* (GA). A escolha desse software se justifica pela sua robustez e eficácia comprovadas ao longo do tempo, permitindo à empresa realizar análises precisas e detalhadas dos seus processos produtivos.

O foco principal deste trabalho recai sobre um dos principais assuntos da empresa, o posicionamento dos sensores responsáveis pela tomada de decisão e controle da linha de produção, sendo assim pretende-se não apenas encontrar a configuração ótima para os sensores, mas também aprimorar a eficiência operacional da indústria como um todo.

1.1. JUSTIFICATIVA E RELEVÂNCIA

A simulação computacional tem sido cada vez mais utilizada para auxiliar na tomada de decisões em diferentes setores (BANKS *et al.*, 2004). Essa técnica de pesquisa é valorizada por sua versatilidade, flexibilidade e poder de análise. No entanto, uma das limitações da simulação

é que ela não é, por si só, uma técnica de otimização. Isso significa que os analistas precisam simular várias configurações do sistema real e selecionar a que apresenta o melhor desempenho.

No entanto, devido ao avanço da tecnologia, com computadores mais rápidos e melhorias nas técnicas de busca e otimização, essa situação vem mudando. Atualmente, é possível combinar a simulação com técnicas de otimização, o que tem levado a um crescimento contínuo do uso dessas abordagens em conjunto (BANKS *et al.*, 2004). Pacotes de simulação disponíveis no mercado já incluem rotinas de otimização integradas, o que facilita a aplicação desses métodos combinados, assim como o GA.

O GA introduzido por Holland (HOLLAND, 1992) vem sendo usado com sucesso para encontrar soluções viáveis para problemas de otimização (GEN; CHENG, 1997). Otimização de processos é um desafio constante em diversas indústrias. Para auxiliar nessa tarefa, o Plant Simulation oferece um pacote de otimização acessível através do “*GAWizard*”. Essa ferramenta facilita a combinação do GA com o ambiente de simulação de eventos discretos (SED), tornando o processo de integração mais rápido e eficiente.

Sendo assim, a aplicação do GA no modelo de simulação fornecerá uma abordagem mais eficiente e sistemática para a busca por solução de um dos problemas atuais da linha, a otimização da configuração dos sensores nas esteiras da linha de produção. Por meio da combinação da simulação computacional com a técnica de otimização, espera-se obter resultados superiores em termos de eficiência, tempo e qualidade na linha de produção de latas.

1.2. OBJETIVO

O objetivo deste trabalho é encontrar um conjunto ideal de posições dos sensores responsáveis por acionar a trava e destrava da entrada de latas nas máquinas da linha de produção. Esse processo será realizado através do uso de algoritmo genético com hiperparâmetros otimizados, com o intuito de maximizar o valor de produção dentro do modelo de simulação no menor tempo possível.

Objetivos específicos:

- Realizar um levantamento de restrições e requisitos do posicionamento dos sensores;

- Corrigir ou minimizar erros relacionados a distribuição do tempo de falha, presentes no atual modelo de simulação;
- Otimizar posicionamento dos sensores com o uso do GA;
- Realizar experimentos e análises comparativas entre a melhor solução encontrada e o modelo atual para avaliar eficácia da solução encontrada;
- Propor uma solução para o chão de fábrica e analisar se o resultado sugerido satisfaz as condições estabelecidas anteriormente;

1.3. CONTEXTO E DESCRIÇÃO DO PROBLEMA

1.3.1. Contexto do problema

A empresa em questão, cujos nomes e dados foram ocultados por questões de confidencialidade, destaca-se no mercado não apenas pela fabricação de embalagens para bebidas, mas também pela sua ampla atuação na produção de embalagens para itens de cuidados pessoais e produtos domésticos e outras tecnologias e serviços para clientes comerciais e governamentais.

A empresa possui uma cultura de inovação, incentivando seus funcionários a identificarem problemas e oportunidades, buscando soluções criativas. A organização também está envolvida em iniciativas de transformação digital, adquirindo aplicativos como Qlik Sense e Tecnomatix Plant Simulation para impulsionar sua eficiência e aprimorar seus processos. Além disso, a empresa está comprometida com a sustentabilidade e tem sido reconhecida por suas práticas e conquistas nesse campo. Através do uso de tecnologia, a organização está impulsionando a inovação em seus produtos e processos, buscando continuamente aumentar sua eficiência energética e produtiva, reduzir emissões de carbono e promover a economia circular.

O problema abordado neste estudo está relacionado à área de inovação e tecnologia, que desempenham um papel fundamental na busca por soluções de melhoria. No entanto, atualmente, a empresa enfrenta dificuldades em propor e implementar novas soluções de otimização da produção no chão de fábrica. Apesar de a empresa possuir uma meta realista por ano, o fato de ser líder de mercado e possuir processos avançados torna desafiador alcançar os valores estabelecidos. Diante desse contexto, a empresa está buscando explorar projetos de

curto e longo prazo da indústria 4.0 como uma estratégia para manter seu ritmo de evolução constante e ascendente.

A indústria 4.0 refere-se à aplicação de tecnologias digitais e inteligentes na cadeia de produção, com o objetivo de aumentar a eficiência, melhorar a qualidade, reduzir custos e promover a inovação (TORTORELLA; FETTERMANN, 2017). Essa abordagem envolve a internet das coisas, big data, inteligência artificial, simulação e outras tecnologias disruptivas. Portanto, a empresa está apostando em projetos que envolvem a implementação dessas tecnologias para impulsionar sua capacidade de otimização da produção.

Após anos de utilização do modelo de simulação no dia a dia dos operadores a busca por novas soluções deixou de ser trivial, abrindo espaço para formas mais robustas na busca de melhorias. Com isso fez-se necessário tomar a decisão de explorar o software de simulação utilizado pela equipe de inovação, que por sua vez foi constatado a disponibilidade de vários pacotes extras disponíveis, porém nunca explorados. Após um estudo abrangente de todos esses pacotes, o GA despertou atenção por sua adequação aos desafios de teste enfrentados naquele momento. Assim, foi decidido dedicar tempo para aprender detalhadamente como utilizar o GA e testar sua capacidade nesse contexto. No entanto, a empresa enfrenta desafios ao tentar equilibrar os costumes tradicionais do chão de fábrica com as oportunidades proporcionadas pela indústria 4.0. Isso pode resultar em dificuldades na implementação de soluções inovadoras, devido à necessidade de altos investimentos e a dificuldade de usar a real capacidade das ferramentas disponíveis por parte dos gestores e supervisores de produção. Portanto, também é objetivo desta pesquisa analisar o uso de novas técnicas tecnológicas para ampliar o conjunto de ferramentas disponíveis na busca pela otimização da produção e aproveitar os benefícios oferecidos.

Como resultado esperado no futuro, almeja-se a otimização do modelo de simulação por meio do uso do GA, o ideal é que os resultados positivos alcançados sejam refletidos no chão de fábrica, estes resultados serão garantidos ao validar o modelo de simulação. Uma vez que esse feito foi alcançado, o GA poderá ser aplicado em outros problemas relacionados à otimização combinatória, aumentando a eficácia do time de inovação. Além disso, ao explorar e utilizar os pacotes oferecidos pelo Tecnomatix Plant Simulation, espera-se comprovar a eficácia dessas ferramentas, abrindo caminho para a adoção de outros pacotes que também possam trazer resultados positivos para o modelo e, conseqüentemente, para a linha de produção.

1.3.2. Descrição do problema

O problema central abordado neste trabalho está relacionado à limitação do uso básico dos operadores responsáveis pela ferramenta de simulação. Atualmente, a empresa utiliza um modelo de simulação para avaliar os impactos das mudanças na configuração da linha de produção e quantificar os benefícios resultantes. No entanto, esse processo é conduzido através de mudanças testadas individualmente. Embora esse método seja aceitável para mudanças simples, como adicionar uma máquina ou trocar uma lógica, quando há múltiplas variáveis envolvidas, o número de testes aumenta exponencialmente.

Atualmente os operadores utilizam a ferramenta "*Experiment Manager*" para solucionar problemas mais complexos. Essa ferramenta calcula todas as possíveis soluções de seu experimento, de forma exaustiva, que por sua vez não é uma boa prática para encontrar um resultado otimizado. Por exemplo a quantidade de simulações executadas para encontrar uma solução para o posicionamento ideal de sensores por esta ferramenta é dada por:

$$\prod_i^n X_i \quad (1)$$

Sendo X_i a quantidade de possíveis posições ocupadas pelo sensor i . Em um cenário geral, se assumirmos que os 40 sensores possuem 100 posições possíveis, tem-se 40^{100} testes, sendo que normalmente é gasto em média 3 minutos de simulação para cada teste, o que totaliza anos de simulação. Por isso, essa prática se torna inviável para o dia a dia.

Nesse contexto, nesse contexto pode-se observar a viabilidade ao uso de otimização combinatória, como heurísticas, para facilitar a busca por resultados próximos ao ótimo, dentro de um tempo razoável. Atualmente, não se usa do conhecimento sobre as técnicas de otimização combinatória na simulação da empresa, o que resulta em um tempo excessivamente longo para resolver problemas ou testar hipóteses na linha de produção. Essa situação compromete a eficiência do processo de inovação por meio da simulação, tendo em vista a necessidade de um alto tempo necessário para desfrutar de seus resultados.

O uso de heurísticas para buscar soluções complexas pode ser aplicado a diversos problemas (ELHOUCINE OUASSAM *et al.*, 2021), incluindo aqueles encontrados no modelo de simulação atualmente utilizado. Um dos problemas frequentemente discutidos refere-se à determinação da posição ideal de todos os pontos de decisões responsáveis pelo controle da linha de produção da fábrica de latas. Devido ao atual comportamento da linha de produção, é

necessário acionar os sensores para controlar a linha e evitar acúmulos e acidentes. Nesse caso específico, a linha de produção conta com mais de diversos sensores responsáveis por modulação das máquinas, sendo que apenas uma parte deles são utilizados para a tomada de decisão dentro do modelo de simulação. A posição individual de cada sensor pode interferir na posição ideal de outros sensores. Portanto, a otimização da disposição desses sensores é o primeiro problema a ser abordado para validar a eficiência do GA proposto.

Mesmo ao utilizar um método de otimização combinatória, pode-se esperar um alto tempo necessário para encontrar uma resposta final, dependendo da complexidade do problema e dos recursos computacionais disponíveis. No entanto, no caso do GA, é possível ajustar seus parâmetros de busca de forma a agilizar a obtenção de uma resposta próxima ao ótimo global. Nesse contexto, torna-se essencial otimizar esses parâmetros de busca, a fim de reduzir ao máximo o tempo necessário para obter uma resposta final do método empregado. Isso permitirá tornar a aplicação das técnicas de otimização mais eficiente e viável para solucionar problemas complexos em diversos domínios.

Outro problema investigado neste trabalho foi a modelagem de falhas utilizada atualmente no modelo, tendo em vista que se trata de um procedimento imprescindível para a aderência dos resultados do modelo aos resultados reais. Simular um comportamento diferente do real pode interferir diretamente nos resultados propostos pelo GA. Ao explorar as informações de falhas utilizadas atualmente identificou-se que o atual modelo de simulação carece de informações sobre as falhas de máquinas, fator que pode impactar diretamente no desfecho deste estudo. Diante da disponibilidade dos dados da empresa, esta pesquisa visa não apenas preencher essa lacuna, mas também se propõe a realizar uma análise dos tempos de falhas das máquinas no modelo. Pretende-se, assim, avaliar a discrepância nos resultados obtidos após a inclusão dessas informações, com o propósito de validar a relevância deste estudo para análises futuras e para qualquer outro modelo desenvolvido pela empresa.

Sendo assim, propõe-se a análise da viabilidade de integrar conceitos de otimização combinatória em modelos de simulação para aprimorar processos industriais. Essa abordagem visa avaliar de forma mais eficiente mudanças na configuração da linha de produção, considerando múltiplas variáveis simultaneamente. A utilização de heurísticas possibilitará a busca por soluções próximas ao ótimo de maneira rápida e precisa, visando reduzir significativamente o tempo necessário para resolver problemas e testar hipóteses. O objetivo final é melhorar a eficiência do processo de inovação por meio da simulação.

2. FUNDAMENTAÇÃO TEÓRICA

2.1. SIMULAÇÃO

Podemos encontrar várias definições distintas sobre simulação na literatura, com base nas definições apresentadas por Banks (1998) e Harrel *et al* (2004), podemos definir simulação como: Criação de uma representação artificial de um processo, sistema ou sistema dinâmico do mundo real, utilizando um modelo computacional. Essa representação é observada ao longo do tempo para obter inferências sobre as características e o desempenho do sistema em questão.

A principal função da simulação é a digitalização de processos, por ser uma função ampla possui usabilidade em diversas áreas, segundo Banks *et al* (2004) alguns exemplos de áreas de aplicação são: projeto e operações em manufatura, gerenciamento de projetos e construção civil, logística, *supply chain* e redes de distribuição, administração do sistema de estoques, modelagem de transporte e tráfego, análise de riscos, aplicações médicas, aplicações militares e aplicações nas mais diversas áreas de serviços, tais como: serviços governamentais, bancos, hotelaria, restaurantes, instituições educacionais, entre outros. A seguir, nos próximos tópicos serão abordadas as definições de sistema, modelo e ambiente, conceito necessário para entender mais a fundo sobre simulação.

2.1.1. Sistema

Banks *et al* (2004) define sistema “um grupo de objetos que são unidos em alguma interação regular e interdependência para a realização de algum propósito”. São exemplos de sistemas: sistema de tráfego, sistema policial, sistema bancário e outros.

Os sistemas ainda podem ser considerados como discretos, contínuos ou combinados (LAW; KELTON; KELTON, 2007). De acordo com Eyring *et al.* (2022), eventos contínuos são caracterizados por mudanças contínuas, como o aumento gradual do nível de água em um lago, que pode assumir uma infinidade de estados. Em contrapartida, eventos discretos são aqueles que ocorrem em momentos específicos no tempo e podem ser identificados, definidos e enumerados.

2.1.2. Modelo

De acordo com Hillier e Lieberman (2021) um modelo é uma representação simplificada de um sistema real, que considera apenas os aspectos relevantes para a análise específica em

questão. Um modelo de simulação computacional ainda pode se dividir em quatro definições diferentes, são elas: Estático, representa o sistema sem levar em conta sua variabilidade; dinâmico, representa o sistema a qualquer tempo; determinístico, quando assumem valores exatos e estocástico, assumem diversos valores dentro de uma distribuição de probabilidades (RENATO; BRIGHENTI, 2006).

2.1.3. Ambiente

Durante o período de 1987 a 2008, ocorreu o início do desenvolvimento dos softwares utilizados atualmente, que trouxeram consigo interfaces gráficas interativas para os usuários. Esses softwares incorporaram recursos de animação e outras ferramentas de visualização, marcando assim o Período de Integração de Ambientes, conforme referenciado por (BANKS *et al.*, 2010). Nesse contexto, o termo "ambientes" é utilizado para descrever qualquer interface de simulação utilizada pelos usuários.

2.2. SIMULAÇÃO DE EVENTO DISCRETOS

A SED é uma técnica amplamente utilizada para analisar e modelar sistemas complexos. A SED tem sido amplamente adotada nas últimas décadas, com um crescimento significativo em sua aplicação científica, seu uso permite a análise detalhada de sistemas discretos, fornecendo informações valiosas para a tomada de decisões (GUIMARÃES *et al.*, 2015). Atualmente, são várias as opções de ambientes de simulação, alguns exemplos de *softwares* que facilitam a introdução da simulação através de interfaces *low code* são o Arena, ProModel, Flexsim, Plant Simulation, Anylogic e outros.

O uso do SED permite a reprodução do estudo do sistema em ambiente computacional, facilitando a avaliação de mudanças nos sistemas sem a necessidade de intervenção no ambiente real (FERRO; CORDEIRO; ORDOÑEZ, 2018). O que, por sua vez, proporciona algumas vantagens significativas, eliminando riscos e custos associados à experimentação no mundo real.

Segundo Lee, Bagheri e Kao (2015), a simulação é uma ferramenta destaque para evolução das indústrias, sendo inclusive citada como um dos pilares da Indústria 4.0. A capacidade de simular cenários virtuais complexos, testar e otimizar processos de produção, realizar simulações de fábricas inteligentes e prever o desempenho do sistema são elementos-chave para alcançar os objetivos da Indústria 4.0.

Neste estudo, será empregado o software "Tecnomatix Plant Simulation" como um ambiente de SED. Disponibilizado pela empresa alemã Siemens PLM Software, esse software é amplamente utilizado para modelagem de processos de produção e logística. Ele proporciona a criação de sistemas logísticos digitais realistas, permitindo a otimização do desempenho desses sistemas. O software é uma ferramenta eficaz para aumentar a eficiência de sistemas existentes, otimizar o consumo de recursos, limitar estoques e reduzir o tempo de produção (SIDERSKA, 2016).

2.3. SIMULAÇÃO COM OTIMIZAÇÃO COMBINATÓRIA

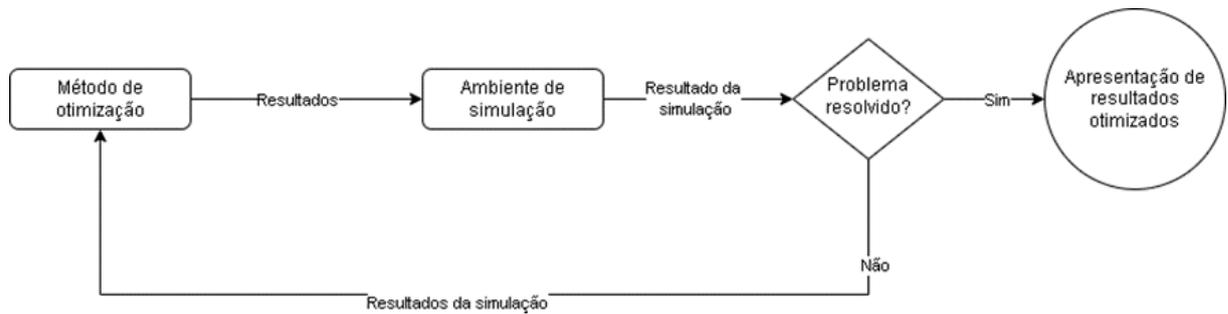
Renato e Brighenti (2006) destacam que a simulação computacional é uma ferramenta importante para prever e visualizar as consequências de diferentes variáveis em um sistema. No entanto, o uso básico dessa técnica dificulta afirmar se os valores encontrados são os ideais para maximizar a produtividade de uma linha de manufatura .

Esta desvantagem pode ser contornada ao utilizarmos os pacotes de otimização que acompanham os softwares, estas ferramentas possuem a capacidade de gerar novas soluções para o problema utilizando heurísticas ou métodos de buscas específicos que por sua vez possuem uma chance maior de obter valores ótimos em troca de alto custo computacional.

Ao aproveitar essas ferramentas, é possível identificar as configurações ideais que maximizam ou minimizam as métricas desejadas. Conforme apontado por Ferro *et al.* (2021), a otimização é um elemento crucial na simulação, permitindo encontrar a melhor solução com base em funções objetivas únicas ou múltiplas, levando em consideração os dados de entrada, restrições e recursos disponíveis.

O papel da otimização então é usar o ambiente para definir os melhores *inputs* que geram os melhores resultados, de acordo com o problema definido na SED. Este ciclo é representado por Fu (2002).

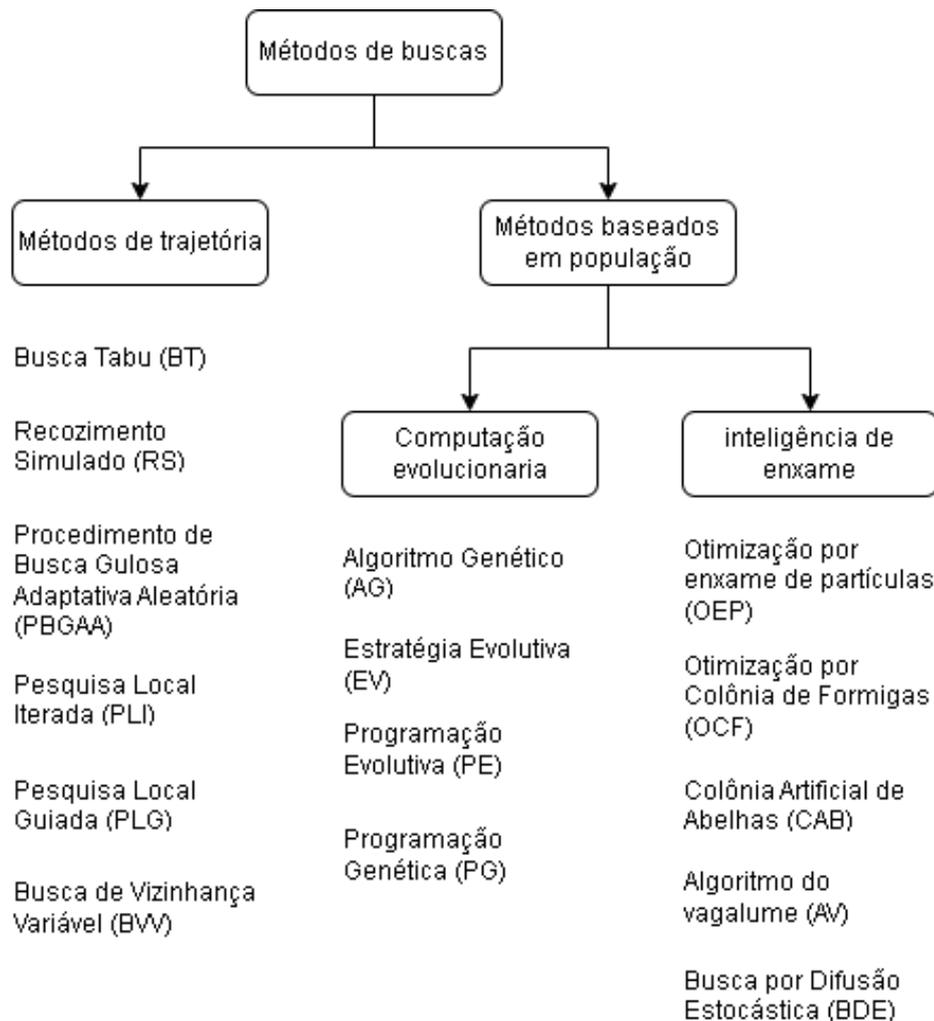
Figura 1 – Fluxograma de otimização em simulação



Fonte: adaptado de FU (2002)

Devemos então escolher a técnica de otimização mais adequada para o problema a ser enfrentado, Azab *et al.* (2021) lista alguns métodos de otimização existentes, para problemas estocásticos as opções mostradas na figura 2.

Figura 2 – Métodos de buscas estocásticas



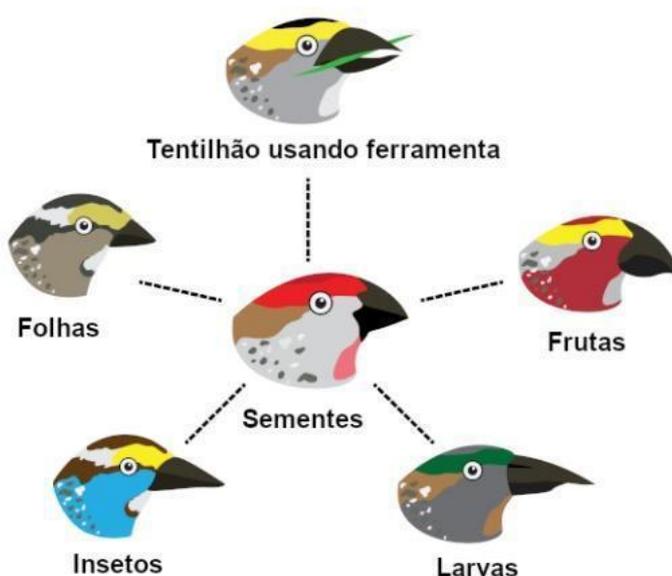
Fonte: adaptado de Azab et al (2021)

Após a exploração da capacidade do Plant Simulation foi percebido a disponibilidade de uma ferramenta que facilita a implementação do GA dentro do modelo de simulação, além de integrar seus resultados ao modelo também permite completa configuração de seus parâmetros de busca e por se tratar de uma ferramenta viável para este tipo de problema foi escolhido para o caso de estudo

2.4. ALGORITMO GENÉTICO

De acordo com Grznár *et al* (2021), os fundamentos dos algoritmos genéticos compartilham os mesmos princípios da biologia. A biologia demonstra no ambiente real, por exemplo, os pássaros que nasciam com o bico que mais facilitava a alimentação em seu ambiente tendiam a viver por mais tempo, neste caso o formato do bico se comporta como uma solução do problema e tem suas distinções representadas na figura 3, que por sua vez cada um irá retornar em um número de anos de vida esperada para o animal, aqueles que vivem mais irão procriar a espécie e passar estas características para os filhos, que vão continuar o ciclo. Neste caso o conjunto de posições de sensores sofrerão alterações com base em um objetivo, a maximização da produção.

Figura 3 – Possíveis bicos de pássaros



Fonte: <https://brasilecola.uol.com.br/biologia/selecao-natural.htm> , acesso em 02/03/2024

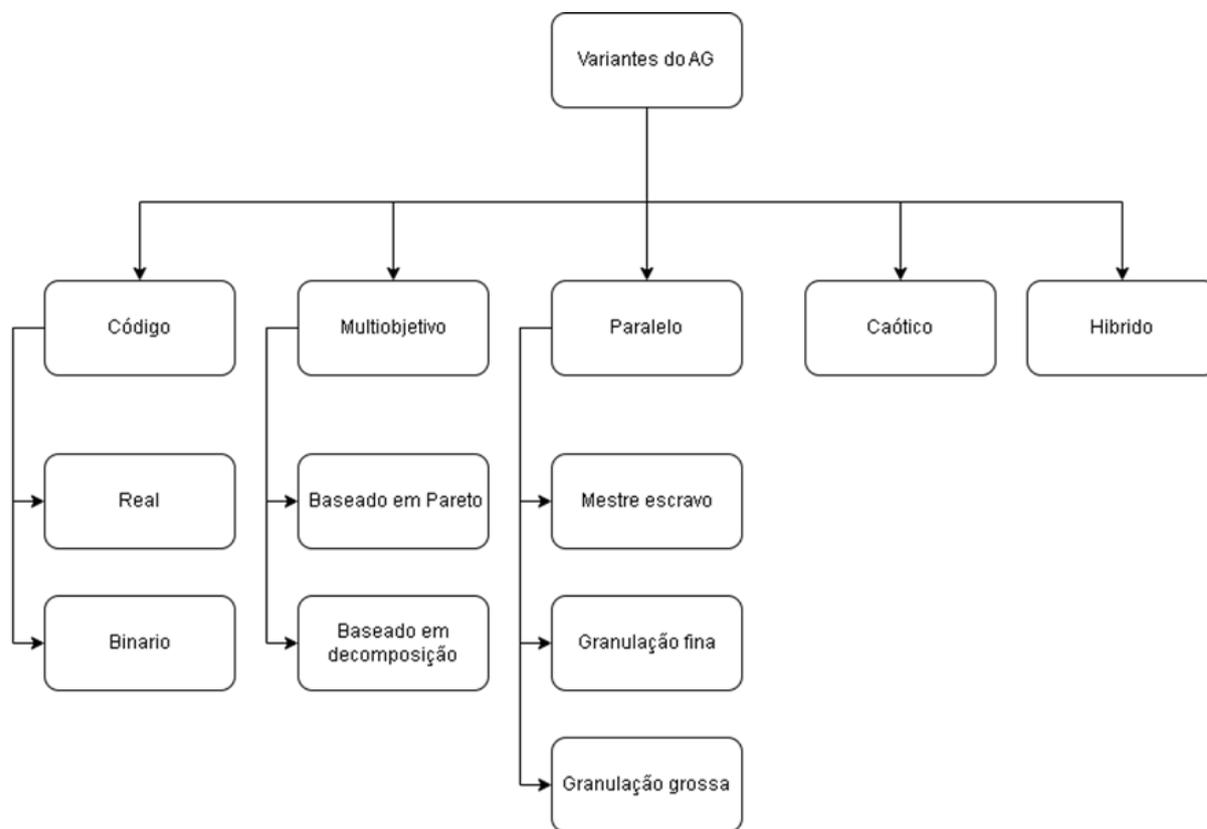
Antes de detalhar cada etapa principal do GA é preciso definir primeiro suas terminologias, Alhijawi e Awajan (2023) as descreve da seguinte forma:

- Cromossomo: contém uma solução na forma de genes;
- Gene: parte de um cromossomo que contém parte de uma solução;
- Indivíduo: sinônimo de cromossomo;
- População: conjunto de cromossomos;
- Avaliação: fornece o mecanismo para avaliar cada indivíduo, através do valor aptidão;
- Critério de parada: critério verificado após cada geração para determinar se deve continuar ou parar de procurar.

Segundo Arabi (2016), o Algoritmo Genético inicia-se com a criação aleatória de uma população de indivíduos. Em cada ciclo do algoritmo, a capacidade de cada indivíduo é avaliada, alguns são escolhidos com base em sua eficácia e submetidos a modificações, incluindo recombinação e possíveis mutações aleatórias, para formar uma nova geração. Essa nova geração é então utilizada na próxima iteração do algoritmo.

Vale salientar que o processo descrito anteriormente é referente ao tipo clássico de GA, existem 5 variantes do GA, são diversas abordagens diferenciadas inspiradas na seleção natural. Suas variantes estão ilustradas na figura 4. Cada variante busca otimizar o desempenho do GA em diferentes contextos e oferecer soluções eficientes para diversos problemas (KATOCH; CHAUHAN; KUMAR, 2021).

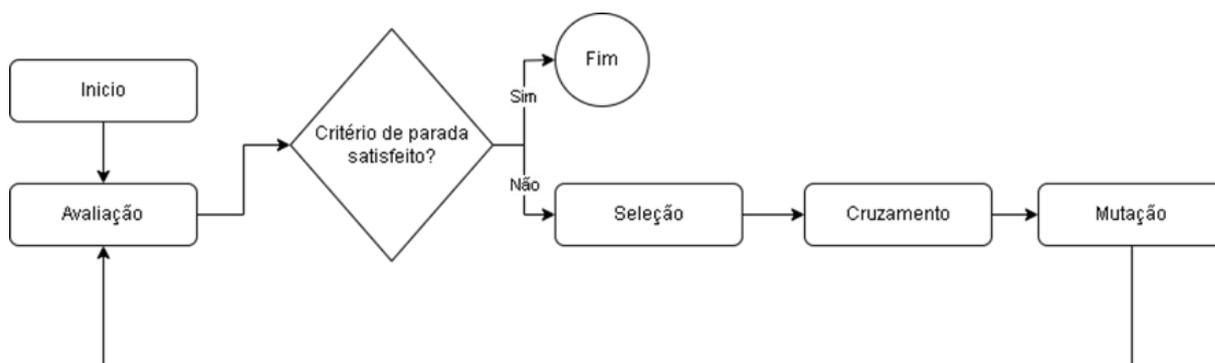
Figura 4 – Listagem de variantes do GA



Fonte: Adaptado de Katoch, Chauhan e Kumar (2021)

Uma das técnicas de otimização estocástica mais fundamentais, com notáveis características evolutivas, é o GA. Esse método utiliza de forma inteligente a seleção aleatória dentro de um espaço de busca predefinido para resolver um problema (GRZNÁR *et al.*, 2021). Este processo é realizado via três processos, seleção, cruzamento e mutação, esquematizados na figura 5.

Figura 5 – Fluxograma do algoritmo genético

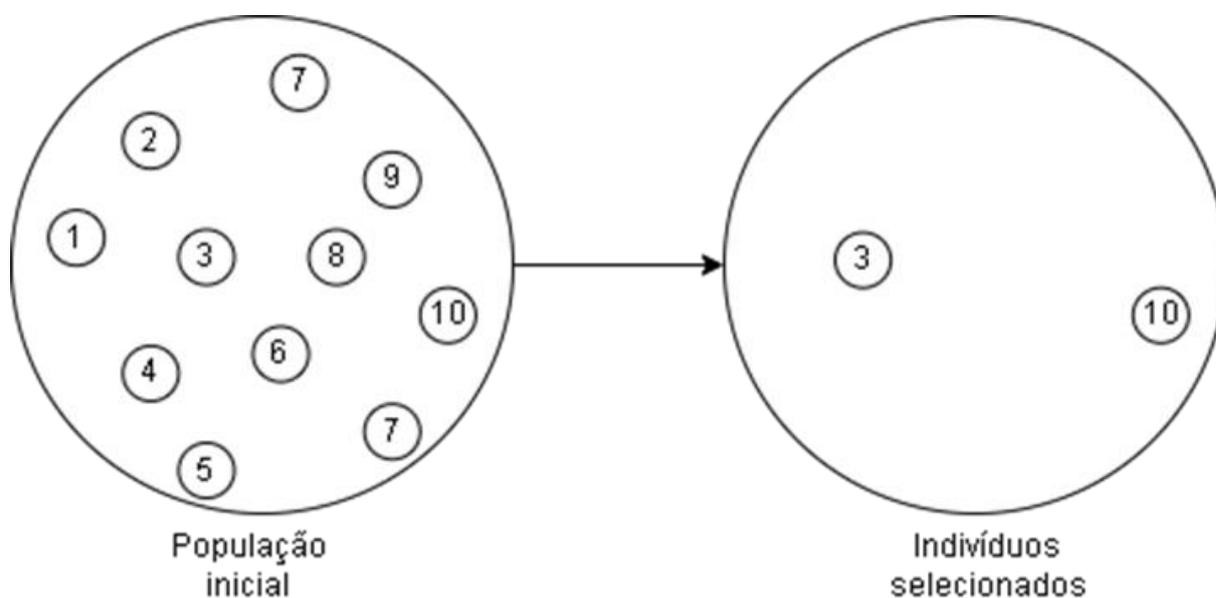


Fonte: Elaboração própria

2.4.1. Seleção

O objetivo da fase de seleção é escolher os indivíduos mais capazes, ou seja, aqueles com maior retorno da função objetivo, e permiti-los passar seus genes para as gerações futuras (MITCHELL, 1996), este processo é representado na figura 6. Os genes com menor retorno de valor aptidão são descartados. Dando início a criação da população que será usada na próxima iteração da heurística.

Figura 6 – Representação de seleção do GA



Fonte: Elaboração própria

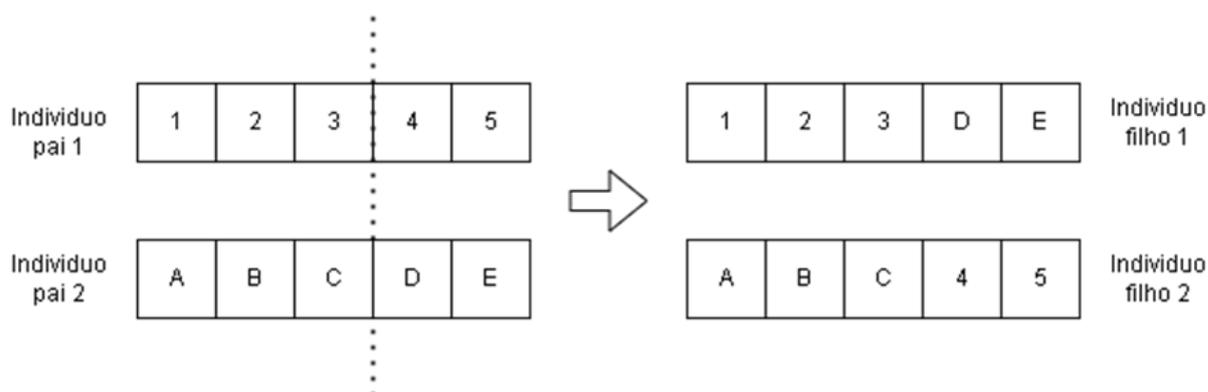
É possível sempre adaptar como este operador genético irá ser executado, quando modificações nestas etapas são verificadas é utilizado o termo algoritmo genético adaptado. Algumas adaptações na fase de seleção são conhecidas como: seleção da roleta, seleção de classificação, seleção de torneio, seleção de Boltzmann e amostragem universal estocástica (KATOCH; CHAUHAN; KUMAR, 2021).

2.4.2. Recombinação

Esta etapa é responsável por combinar informações genéticas de dois indivíduos selecionados para criar descendentes. Durante a recombinação, os pontos de corte são determinados aleatoriamente nos cromossomos dos pais, e as partes correspondentes são

trocadas entre eles, este processo é representado na figura 7. Essa troca de material genético permite a criação de descendentes que possuem características herdadas de ambos os pais (GOLDBERG, 1989). O processo de recombinação promove a diversidade genética na população, explorando diferentes combinações genéticas e aumentando a chance de encontrar soluções melhores.

Figura 7 – Representação da recombinação do GA



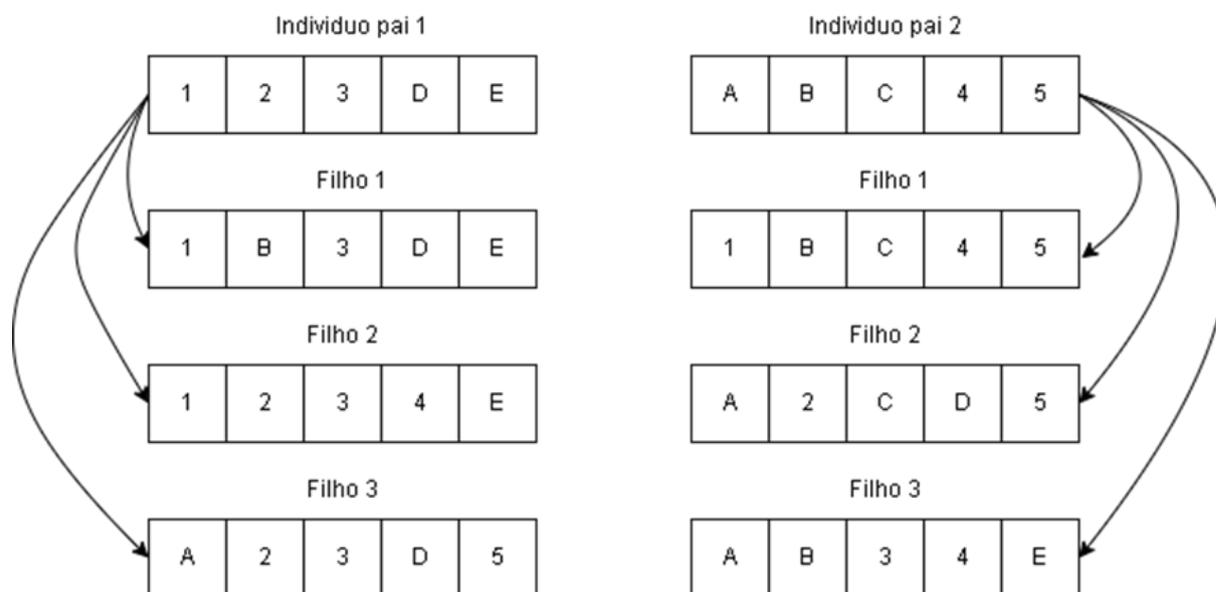
Fonte: Elaboração própria

Também é notável a existência de operadores de recombinação adaptados para o uso do algoritmo genético. Algumas existentes são: Cruzamento de ponto único, Cruzamento do ponto K, cruzamento uniforme, Cruzamento parcialmente mapeado, Cruzamento de preservação de precedência, Cruzamento aleatório, Cruzamento reduzido, Cruzamento de ciclo (KATOCH; CHAUHAN; KUMAR, 2021). A ferramenta disponível no Plant simulation utiliza por padrão o cruzamento de ponto único em seus procedimentos

2.4.3. Mutação

A operação de mutação é empregada para ampliar a exploração do espaço de soluções e evitar que o GA convirja prematuramente para ótimos locais. A mutação consiste em modificar o valor de um gene em um indivíduo selecionado aleatoriamente, este processo é representado na figura 8, com uma determinada probabilidade conhecida como probabilidade de mutação (BODENHOFER, 2004). Dessa forma, diversos indivíduos na nova população podem ter um de seus genes alterado de maneira aleatória.

Figura 8 – Representação da mutação do GA



Fonte: Elaboração própria

Katoch, Chauhan e Kumar (2021) listam alguns operadores de mutação adaptados, são eles: Mutação de deslocamento, Mutação de inversão, Mutação embaralhada, Mutação de inversão de bits, Mutação reversa.

2.4.4. Parâmetros do algoritmo genético

Os parâmetros do algoritmo genético são fundamentais para o sucesso da aplicação desta técnica. Eles são responsáveis por determinar a eficiência e a eficácia do algoritmo na solução do problema proposto. São eles:

- Taxa de recombinação: A taxa de recombinação, representa o número de vezes que o cruzamento ocorre entre cromossomos em uma geração, determina a probabilidade de que dois cromossomos troquem algumas de suas partes. Uma taxa de cruzamento de 100% indica que todos os descendentes são gerados por meio do cruzamento. Por outro lado, uma taxa de 0% implica que a nova geração será idêntica à geração anterior, exceto pelos indivíduos resultantes do processo de mutação. A taxa de cruzamento varia dentro da faixa de [0, 1] (DE JONG; SPEARS, 1992).
- Taxa de mutação: Essa taxa define a quantidade de cromossomos que serão mutados em uma geração; a taxa de mutação varia entre [0,1]. Em casos em

que a mutação ocorre com muita frequência, o GA pode se tornar uma busca aleatória descontrolada (OBITKO, 1998).

- Tamanho da população: O tamanho da população indica o número total de indivíduos que compõem a população. A seleção do tamanho adequado é uma questão delicada; se o tamanho da população é pequeno, significa que o espaço de busca disponível é limitado, o que pode levar ao alcance de ótimos locais. No entanto, se o tamanho da população for muito grande, a área de busca é aumentada e a carga computacional torna-se elevada (ROEVA; FIDANOVA; PAPRZYCKI, 2013).
- Número de gerações: Essa medida se refere ao número de ciclos antes da conclusão do algoritmo. Em algumas situações, algumas centenas de iterações são suficientes, enquanto em outros casos podem ser necessário um número maior, tudo dependendo do tipo e complexidade do problema. A utilização desse parâmetro pode variar de acordo com o projeto do GA, e em certos casos ele pode não ser utilizado, especialmente quando o término do GA depende de critérios específicos em vez de um número fixo de iterações (HASSANAT *et al.*, 2019).

2.5. MODELAGEM DE FALHAS

Analisar o tempo de falha da máquina é crucial para otimizar os processos industriais, isto permite a detecção e previsão de falhas antes que elas ocorram, permitindo a manutenção proativa e minimizando o tempo de inatividade. Ao identificar possíveis anomalias e falhas com antecedência, as empresas podem substituir elementos desgastados e otimizar os cronogramas de manutenção, reduzindo custos e melhorando o desempenho geral (HASSAN; SADEK; ATTIA, 2023). Além disso, compreender o impacto das falhas na capacidade de produção é essencial para o gerenciamento da produção, pois ajuda a identificar gargalos e priorizar subsistemas para melhoria.

O tempo de falha das máquinas no sistema é frequentemente descrito por meio de modelagem como variáveis aleatórias, visando incorporar as incertezas provenientes de erros de medição, ruídos ou pequenas perturbações nos processos de fabricação. Ao introduzir uma variável aleatória como entrada, a saída do modelo também se transforma em uma variável aleatória (PEHERSTORFER; KRAMER; WILLCOX, 2017). Para que essa saída aleatória

reflita com precisão a realidade, é necessário ajustar os dados de falha da máquina a uma distribuição de probabilidade específica, com parâmetros definidos. Esse processo permite modelar um comportamento similar ao real da máquina dentro do sistema (LEAL; ANDRADE, 2018).

As duas métricas utilizadas neste trabalho são o MTTR (*Mean Time to Repair*) e o MTBF (*Mean Time Between Failures*), Sellitto (2007) por sua vez as define que MTBF representa a estimativa do período médio de operação de um componente ou sistema até que ocorra uma falha, enquanto o MTTR, por sua vez, é o período médio necessário para realizar a restauração de um sistema ou componente após a ocorrência de uma falha

Um dos métodos empregados na condução de análises de confiabilidade e disponibilidade é o método quantitativo (RÉGIS ANDRÉ WUTTKE; MIGUEL AFONSO SELLITTO, 2008). Neste contexto, a modelagem de falhas é estabelecida através da análise de dados históricos dos tempos entre falhas, os quais são modelados utilizando distribuições de probabilidade, com o suporte de software estatístico ou programação. A partir da análise do comportamento dos modos de falhas de um determinado tipo de sistema ao longo do tempo, é possível associar distribuições de probabilidades aos tempos entre falha e de reparo do sistema. Esse procedimento permite uma compreensão mais profunda da performance do sistema ao longo do tempo e facilita a previsão de sua confiabilidade e disponibilidade futuras.

Para garantir que esta modelagem esteja correta, são necessários métodos estatísticos para verificar se a hipótese de que o conjunto de dados se assemelha à distribuição de probabilidade e aos parâmetros sugeridos. Entre os testes de aderência mais conhecidos, destacam-se o teste *Qui-Quadrado (QQ)*, o teste de *Kolmogorov-Smirnov (KS)* e o teste de *Anderson-Darling (AD)* (CLAESKENS; HJORT, 2008).

Além desses testes de aderência, outro conceito essencial na análise estatística é o de verossimilhança, sua função é indicar o quão provável é que os parâmetros do modelo tenham gerado os dados observados (CASELLA; BERGER, 2017).

Para modelos probabilísticos, como os utilizados na modelagem de confiabilidade e disponibilidade, a verossimilhança é frequentemente calculada como a probabilidade conjunta dos dados observados sob o modelo especificado. O objetivo é maximizar a verossimilhança, ou seja, encontrar os valores dos parâmetros que tornam os dados observados mais prováveis sob o modelo (CASELLA; BERGER, 2017).

Assim, ao realizar testes de aderência, como os testes QQ, KS e AD, este trabalho avalia não apenas a adequação da distribuição de probabilidade proposta aos dados observados, mas também a plausibilidade dos parâmetros do modelo estimados com base na verossimilhança. Esses testes combinados permitem uma análise mais completa e crítica da relação entre o modelo estatístico e os dados observados, possibilitando inferências mais confiáveis sobre o sistema em estudo (CLAESKENS; HJORT, 2008).

3. REVISÃO DE LITERATURA

Há bastante tempo, a SED tem sido uma das ferramentas mais amplamente utilizadas para auxiliar na tomada de decisões organizacionais. Trabalhos, como Villagomez *et al.*, (2019) destacam que o SED é uma ferramenta essencial para a tomada de decisões eficazes sem a necessidade de implementação física ou custos adicionais com testes reais. O estudo mencionado, em particular, utiliza as ferramentas de simulação para otimizar o uso de matérias-primas na produção de peças de veículos, mas o escopo de aplicação do SED vai além disso.

Diversas outras literaturas enfatizam o amplo uso dos conceitos do SED. Harrison *et al* (2007) afirma que as técnicas de SED possuem múltiplas finalidades, incluindo predição, comprovação, explicação, prescrição e orientação empírica. Essa diversidade de aplicação destaca a versatilidade e abrangência dessas técnicas em diversos campos de estudo e tomada de decisão.

Algumas aplicações industriais exemplares demonstram o uso efetivo da simulação em diferentes contextos, Tao *et al* (2019), exemplifica o uso de simulação por empresas como Siemens, General Electric e Airbus aplicam a simulação em diferentes áreas como planejamento, operação e manutenção de sistemas, otimização de ativo e dos processos de produção. Esses exemplos ilustram como a simulação desempenha um papel crucial na indústria, impulsionando melhorias e resultados positivos em diversas áreas, além de se provar bastante versátil e útil para a resolução de problemas e otimização de empresas.

O GA por si só também há bastante usabilidade na literatura, de acordo com Alam *et al* (2020) os algoritmos já foram utilizados em aplicações de otimização multiparâmetros, classificação, evolução de programas de computador e estruturas computacionais, em aprendizado de máquina, modelagem de mudança organizacional e simulação de sistemas cognitivos. Provando que também é um conceito versátil com bastante potencial de uso.

Em relação ao GA para resolução de problemas de otimização combinatória, na literatura seu uso também é bastante notável, até mesmo em conjuntos com alguns ambientes de simulação, em vários tipos de indústrias. Miranda, Montevechi e de Pinho (2015) afirmam que o GA proposto para otimização de modelos de SED obteve resultados satisfatórios em relação à qualidade de resposta e eficiência.

Também foi possível notar o uso do GA para determinar a posição ideal de sensores, De acordo com Soubhia e Serpa (2018), os algoritmos genéticos têm sido amplamente utilizados para solucionar problemas de posicionamento de atuadores e sensores, tanto em situações contínuas quanto discretas. Também é citado outros autores como o (CHAKRABORTY; RATHORE; ROY, 2012) que utilizaram deste algoritmo para resolver problemas similares, em situações diferentes, porém, com o mesmo objetivo, maximizar o resultado do sistema.

Por mais que o GA seja um conceito relativamente antigo, é possível notar recentemente diversos trabalhos com esta temática e evolução deste método, o que comprova sua relevância. (ALHIJAWI; AWAJAN, 2023) resume a última década de pesquisa com temas relacionados ao GA, mostrando que o espaço de áreas de atuação é ainda maior, com adição de áreas como processamento de imagem, escalonamento, agrupamento, engenharia de software, processamento de linguagem natural e sistema de recomendação.

Em outro artigo relevante Wang, Xu e Wang (2023) apresenta adaptações no GA visando melhorar seu desempenho. Essas adaptações envolvem duas etapas de recombinação e mutação em conjunto, resultando em uma maior eficiência do GA. Essas práticas de melhoria, adaptação ou extensão dos operadores genéticos são comuns na literatura e levam ao surgimento de novas versões do GA, tão relevantes quanto à inicial para resolver problemas complexos. Isso demonstra a utilidade e versatilidade do método do GA.

Uzuner Şahin *et al* (2024) destacam a importância de considerar a reparabilidade dos componentes, demonstrando que a inclusão desse fator pode resultar em ganhos significativos, como um aumento de 3% na confiabilidade do sistema. Além disso, destaca a relevância de abordagens inovadoras, como algoritmos genéticos e simulação de Monte Carlo, para aprimorar e alcançar sistemas mais confiáveis e eficientes.

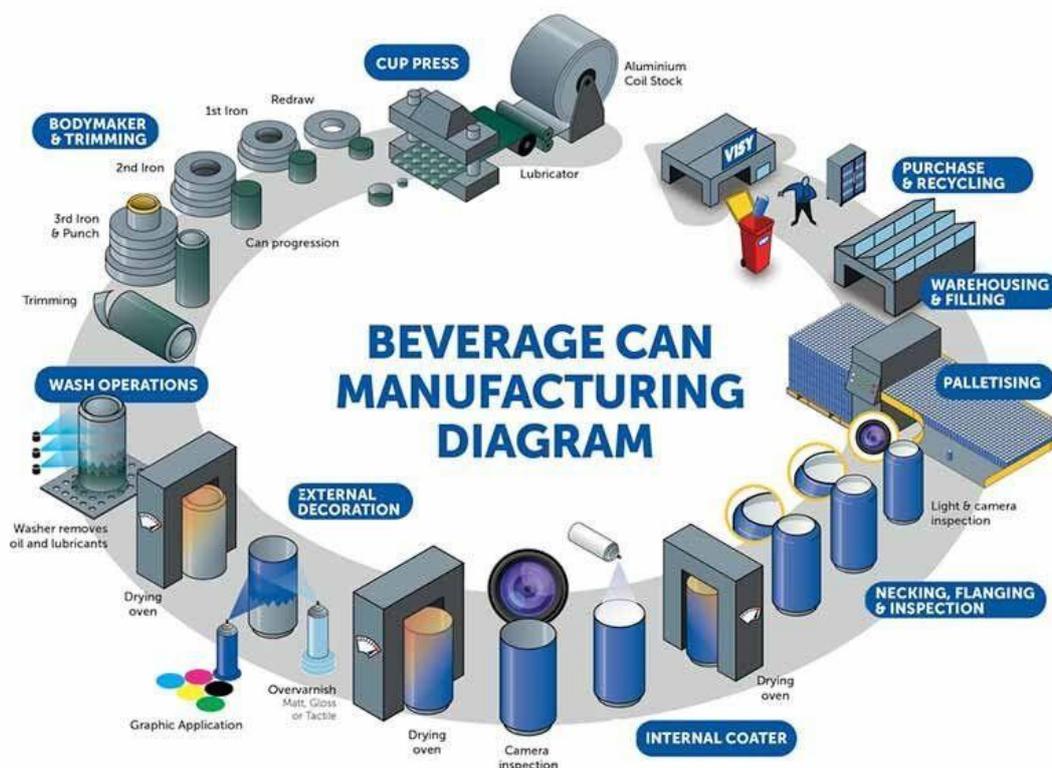
Por fim, Albus *et al* (2024) abordam a otimização de recursos em linhas de montagem robóticas usando um algoritmo genético modificado, provando sua grande relevância nos dias de hoje. O estudo revela melhorias na escalabilidade do algoritmo, embora os valores objetivos tenham sido ligeiramente inferiores. O trabalho optou pela utilização de clusters de atividades e a fusão desses clusters com base em restrições de precedência. Conclui-se que o algoritmo genético mostra potencial de aplicação em indústrias para reconfigurar as linhas de montagem, mas são necessárias otimizações adicionais para aproveitar todo o seu potencial.

4. ESTUDO DE CASO

4.1. APRESENTAÇÃO DO MODELO

Dentro da fábrica ocorrem os processos necessários para transformar a folha de alumínio em latas, dos mais diversos tamanhos e formatos, no geral o alumínio passa por procedimentos como demarcação, corte, lavagem, pintura, envernizamento, criação de pescoço, checagem de conformidade e pôr fim a paletização das latas. A visão micro do processo é representada na figura 9, mostrando todas as transformações que o alumínio passa para formar uma lata através das máquinas mencionadas.

Figura 9 – Representação do Fluxo de atividades da lata

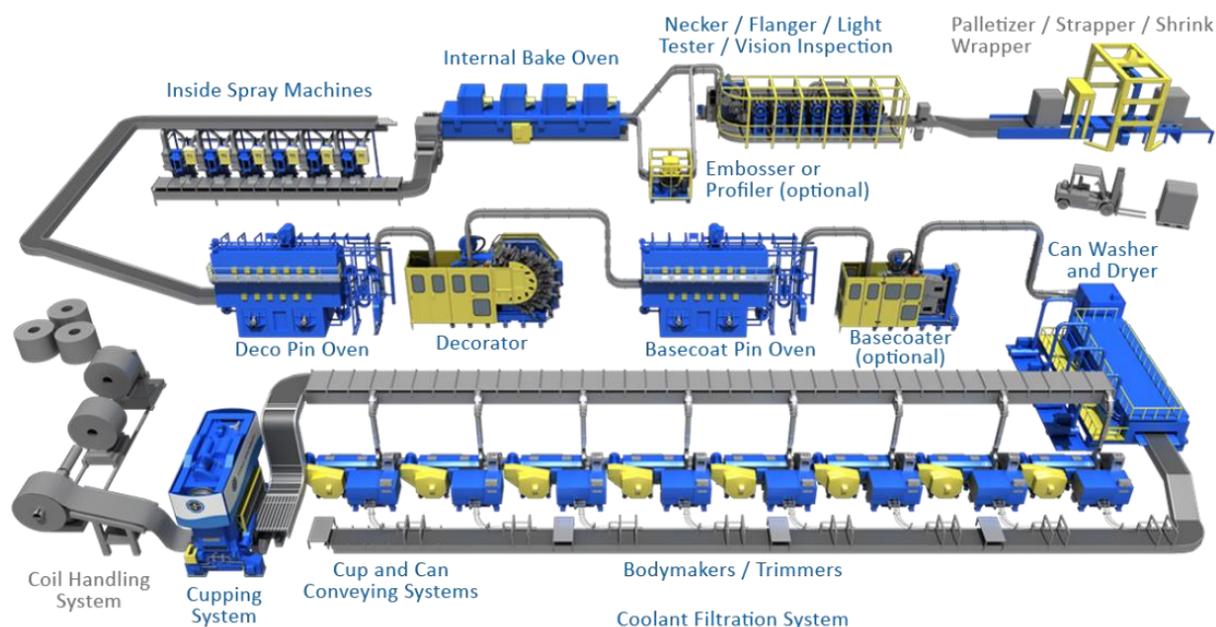


Fonte: <https://www.biopineoe.com/the-new-beverage-can-aluminum-250ml-slim-can-oem>, acesso em 21/02/2023

Já o modelo de simulação é formado por 25 máquinas, sendo 1 *Cupper* (CUP), 8 *Body Makers* (BM), 1 *Washer* (WSH), 2 *Printers* (PRT), 8 *Inside Spray* (IS), 1 *Internal Bake Oven* (IBO), 1 *Necker* (NCK), 2 *Light Tester* (LT) e 1 *Palletizer* (PLT). Todas elas possuem esteiras com entre si encarregadas de fazer a movimentação das latas. A figura 10 representa bem o

fluxo de processos dessas máquinas no chão de fábrica. A seguir será detalhada o funcionamento de cada uma.

Figura 10 – Representação da linha de produção



Fonte: <https://www.stollemachinery.com/pt-br/linhas-para-latas-de-bebidas>, acesso em 21/02/2023.

O sistema de produção de copos (CUP) ilustrada na figura 11, é uma prensa utilizada na indústria para estampar ou conformar folhas de alumínio, transformando-as no formato desejado para a produção de latas. Essa etapa é crucial para garantir a consistência e a precisão nas dimensões das latas, bem como para garantir a qualidade do produto final. Suas atividades são geralmente compostas por 4 etapas, são elas:

1. Alimentação de Material: A máquina é carregada com folhas de alumínio, que podem ser bobinas ou folhas cortadas, dependendo do processo adotado pela fábrica.
2. Estampagem: A CUP realiza a estampagem das folhas de alumínio, aplicando pressão controlada para moldar as folhas no formato desejado para as latas. Moldes específicos podem ser utilizados para criar o perfil exato da lata.
3. Corte e Finalização: A máquina pode, em alguns casos, realizar operações adicionais, como o corte de bordas ou outras operações de acabamento, dependendo dos requisitos do processo de produção.

4. Retirada do Produto: Após a estampagem, as folhas agora conformadas assumem a forma da base das latas e são retiradas da máquina para as etapas subsequentes do processo de fabricação.

Figura 11 – Máquina CUP



Fonte: <https://www.stollemachinery.com/pt-br/linhas-para-latas-de-bebidas>, acesso em 21/02/2023.

As BM, ilustrada na figura 12, são projetadas para realizar a formação do corpo da lata a partir das folhas já processadas pela CUP. Sua função é moldar o corpo da lata. Essas máquinas são projetadas para lidar com uma variedade de tamanhos e formatos de latas, produzindo latas entre 220ml a 473ml.

Figura 12 – Máquina BM



Fonte: <https://www.stollemachinery.com/pt-br/linhas-para-latas-de-bebidas>, acesso em 21/02/2023.

A máquina lavadora de latas (WSH), ilustrada na figura 13, têm a função de remover resíduos, sujeira e quaisquer contaminantes que possam estar presentes nas latas antes do enchimento com produtos para garantir a higiene e a limpeza das latas antes de serem preenchidas com produtos como alimentos ou bebidas. Isso é fundamental para cumprir os padrões de qualidade e segurança alimentar, além de garantir que o produto final esteja livre de impurezas que possam comprometer a qualidade ou a segurança do produto. Podemos descrever as atividades desta máquina em 7 etapas, são elas:

1. Alimentação das Latas: As latas vazias entram na máquina lavadora, em uma esteira transportadora, para iniciar o processo de limpeza.
2. Pré-Lavagem: Antes da lavagem principal, algumas máquinas podem ter uma etapa de pré-lavagem, onde jatos de água são aplicados para remover resíduos grosseiros e partículas soltas.
3. Lavagem Principal: A lavadora de latas utiliza uma combinação de água e detergentes específicos para lavar as latas de maneira eficiente. Jatos de água pressurizada são direcionados para as latas, removendo sujeira, poeira e outros contaminantes.
4. Escovação ou Outros Métodos Mecânicos: Em algumas máquinas, escovas ou métodos mecânicos podem ser usados para garantir uma limpeza mais profunda, especialmente em áreas de difícil acesso.

5. Enxágue: Após a lavagem, as latas passam por um processo de enxágue para remover qualquer resíduo de detergente ou partículas soltas.
6. Secagem: Algumas máquinas podem incluir um processo de secagem, como o uso de ar quente, para garantir que as latas estejam completamente secas antes de prosseguir para as próximas etapas do processo de produção.
7. Saída das Latas Limpas: As latas limpas e secas são então encaminhadas para as etapas subsequentes do processo de produção.

Figura 13 – Máquina WSH



Fonte: <https://www.stollemachinery.com/pt-br/linhas-para-latas-de-bebidas>, acesso em 21/02/2023.

A máquina decoradora (PRT), ilustrada na figura 14, tem como função imprimir designs, logotipos, rótulos, códigos de barras e outras informações gráficas nas latas. Isso não apenas torna as latas visualmente atraentes, mas também fornece informações importantes para os consumidores e facilita o rastreamento de produtos. Podemos descrever as atividades desta máquina em 4 etapas, são elas:

1. Configuração da Máquina: A máquina decoradora é configurada de acordo com o design específico a ser aplicado. Isso pode envolver o carregamento de tintas coloridas ou outros materiais de impressão, bem como a programação da máquina para garantir precisão na aplicação do design.

2. Alimentação das Latas: As latas entram na máquina decoradora geralmente em uma esteira transportadora ou mecanismo semelhante, prontas para receber a impressão.
3. Aplicação da Tinta: A máquina utiliza tecnologia de impressão para aplicar a tinta diretamente nas latas. Isso pode envolver processos como serigrafia, impressão por transferência ou tecnologias de impressão digital, dependendo da máquina e das especificações do processo.

Figura 14 – Máquina PRT



Fonte: <https://www.stollemachinery.com/pt-br/linhas-para-latas-de-bebidas>, acesso em 21/02/2023.

4. Secagem: Após a aplicação da tinta, as latas podem passar por um processo de secagem para garantir que a tinta esteja completamente fixada e que as latas estejam prontas para as etapas subsequentes do processo de produção. Esta etapa é realizada por um forno, ilustrado na figura 15, localizado após a etapa de aplicação do rótulo.

Figura 15 – Máquina forno da PRT



Fonte: <https://www.stollemachinery.com/pt-br/linhas-para-latas-de-bebidas>, acesso em 21/02/2023.

A máquina aplicadora de verniz (IS), ilustrada na figura 16, sua função é aplicar um revestimento interno de verniz nas latas. Esse revestimento atua como uma barreira protetora entre o metal da lata e o conteúdo embalado, impedindo a corrosão e garantindo a qualidade e segurança dos alimentos ou bebidas armazenados. Podemos descrever as atividades desta máquina em 5 etapas, são elas:

1. Preparação da Máquina: Antes de iniciar o processo, a máquina é preparada para garantir que todos os sistemas estejam operacionais. Isso inclui a verificação e o abastecimento dos reservatórios de verniz, a calibração dos bicos de spray e outros ajustes necessários.
2. Alimentação das Latas: As latas vazias são alimentadas na máquina e são posicionadas de maneira que o interior esteja acessível para receber o revestimento de verniz.
3. Aplicação do Verniz: A máquina utiliza um sistema de pulverização para aplicar o verniz no interior das latas. O verniz é pulverizado de maneira uniforme para garantir uma cobertura completa e consistente.
4. Giro das Latas: Em alguns casos, as latas podem ser giradas durante o processo para garantir uma distribuição uniforme do verniz em todas as áreas internas.

5. Saída das Latas Tratadas: As latas, agora revestidas internamente com verniz, estão prontas para as próximas etapas do processo de produção, como a aplicação de designs externos, o enchimento do produto e o fechamento da lata.

Figura 16 – Máquina IS



Fonte: <https://www.stollemachinery.com/pt-br/linhas-para-latas-de-bebidas>, acesso em 21/02/2023.

O Forno de Cura (IBO), ilustrado na figura 17, tem como função principal acelerar o processo de secagem e cura do revestimento aplicado nas latas. A cura é uma etapa crucial para garantir que o revestimento de verniz seque atinja as propriedades desejadas, como durabilidade, aderência, resistência química e outras características importantes para a qualidade do produto final. Podemos descrever as atividades desta máquina em 5 etapas, são elas:

1. Alimentação das Latas: As latas que receberam algum tipo de revestimento, como tinta ou verniz, são alimentadas na entrada do forno de cura.
2. Aquecimento: No interior do forno, as latas são expostas a temperaturas elevadas. A temperatura e o tempo de exposição podem variar dependendo do tipo de revestimento e das especificações do processo.
3. Cura do Revestimento: Durante a permanência no forno, o revestimento aplicado nas latas passa por um processo de cura, que envolve a secagem e a reação química para atingir as propriedades desejadas.

4. Resfriamento: Após o processo de cura, as latas passam por uma zona de resfriamento dentro do forno. Isso é feito para garantir que as latas não saiam do forno a uma temperatura que possa comprometer a qualidade do produto ou causar danos.
5. Saída das Latas Curadas: Após a cura completa, as latas estão prontas para as próximas etapas do processo de produção, como a decoração, o enchimento do produto e o fechamento da lata.

Figura 17 – Máquina IBO



Fonte: <https://www.stollemachinery.com/pt-br/linhas-para-latas-de-bebidas>, acesso em 21/02/2023.

A máquina formadora de pescoço (NCK), ilustrada na figura 18, é projetada para criar a borda ou o pescoço característico na parte superior da lata, onde a tampa será posteriormente selada. Sua função principal é moldar a parte superior da lata, criando uma borda reforçada conhecida como "pescoço". Esse pescoço é fundamental para o processo de vedação posterior, garantindo uma selagem hermética e apropriada da lata. Podemos descrever as atividades desta máquina em 3 etapas, são elas:

1. Alimentação das Latas: As latas vazias são alimentadas na máquina, através de um sistema de transporte, posicionando-as adequadamente para a formação do pescoço.
2. Moldagem do Pescoço: A máquina realiza operações de estampagem e conformação para moldar a parte superior da lata, criando o pescoço reforçado. Esse processo pode envolver o uso de matrizes e ferramentas específicas para alcançar o formato desejado.

3. Saída das Latas Formadas: Após o processo de formação do pescoço, as latas estão prontas para as etapas subsequentes do processo de produção.

Figura 18 – Máquina NCK



Fonte: <https://www.stollemachinery.com/pt-br/linhas-para-latas-de-bebidas>, acesso em 21/02/2023.

O teste de luz (LT), ilustrado na figura 19, tem como função identificar possíveis vazamentos de luz na lata, indicando falhas na selagem. Isso é crucial para garantir a qualidade e a integridade dos produtos armazenados nas latas, especialmente quando se trata de produtos que exigem vedação hermética, como bebidas carbonatadas. Podemos descrever as atividades desta máquina em 5 etapas, são elas:

1. Alimentação das Latas: As latas, após passarem por várias etapas do processo de produção, são alimentadas na máquina LT para a realização do teste de vedação.
2. Pressurização ou Vácuo: A máquina cria um ambiente controlado ao redor da lata, geralmente aplicando pressurização ou vácuo, criando uma condição que possibilita a detecção de eventuais vazamentos de gás ou líquido.
3. Detecção de Luz: Durante o processo, a máquina utiliza um sistema de detecção de luz sensível para verificar se há qualquer vazamento de luz ao redor da lata. Isso pode ser feito através da iluminação da lata e da detecção de qualquer transmissão de luz externa.

4. Registro de falhas: Se a máquina LT detectar vazamentos de luz, ela registra a falha para que as latas com problemas possam ser identificadas e retiradas da linha de produção.
5. Rejeição Automática ou Sinalização: Dependendo da configuração da máquina, as latas com falhas podem ser automaticamente removidas da linha de produção ou podem ser sinalizadas para que os operadores possam realizar uma inspeção mais detalhada.

Figura 19 – Máquina LT



Fonte: https://www.sencon.com/pt/2pc/upgrade_light_tester_package, acesso em 21/02/2023

A paletizadora (PLT), ilustrada na figura 20, é uma máquina utilizada no final do processo de produção de latas, principalmente na etapa de embalagem e preparação para o transporte. Sua função principal é organizar e empilhar as latas de maneira eficiente em paletes. Isso facilita a movimentação das latas dentro da fábrica, a armazenagem em estoque e o transporte para distribuição. Podemos descrever as atividades desta máquina em 6 etapas, são elas:

1. Alimentação das Latas: Após passarem por todas as etapas de produção, as latas são alimentadas na paletizadora por meio de uma esteira transportadora.
2. Arranjo e Formação de Camadas: A paletizadora é programada para criar camadas organizadas de latas no palete. Isso pode envolver a formação de padrões específicos ou a otimização da quantidade de latas em cada camada.

3. **Elevação e Posicionamento:** A máquina eleva cada camada de latas à altura adequada e posiciona-a no palete em construção. Isso pode envolver sistemas de elevação e ventosas ou garras mecânicas para manipular as latas.
4. **Interleaving de Camadas:** Em algumas paletizadoras, as camadas podem ser intercaladas com folhas de papelão ou outro material de separação para facilitar a estabilidade e proteção das latas durante o transporte.
5. **Fixação e Compactação:** Após cada camada ser posicionada, a paletizadora pode aplicar pressão ou utilizar dispositivos para compactar as latas e garantir um empilhamento estável.
6. **Finalização e Envio:** Uma vez que o palete atinge as dimensões desejadas e está completo, a paletizadora pode movê-lo para uma área de finalização, onde é envolvido em filme plástico para estabilização e, em seguida, está pronto para o transporte.

Figura 20 – Máquina PLT



Fonte: <https://www.arrowconveyor.com/product/alpha-turbo-high-level-bulk-depalletizer>, acesso em 21/02/2023

As esteiras transportadoras, ilustrada na figura 21, são situadas entre as máquinas, sua função é o transporte e atuam como um *buffer* de latas em situações de falha das máquinas. Adicionalmente, ao longo de toda a extensão da esteira, sensores são perceptíveis, os quais captam, armazenam e transmitem informações acerca da capacidade atual da esteira. Esses dados são empregados para emitir comandos que controlam a abertura ou fechamento das entradas.

Figura 21 – Esteiras com latas



Fonte: <https://www.stollemachinery.com/pt-br/linhas-para-latas-de-bebidas>, acesso em 21/02/2023.

Além de máquinas e esteiras o modelo é composto de lógicas de programação que consiste em representar as lógicas dos computadores lógicos programáveis (CLP) do chão de fábrica e de sensores ao longo das esteiras, responsáveis obter informações do fluxo de latas para que a lógica de CLP funcione dentro do modelo de simulação. Por fim, há também a existência de grandes buffers, responsáveis por acumular latas entre 2 processos críticos quando necessário, fazendo com que o fluxo de latas seja interrompido apenas em casos graves de falhas de máquinas.

4.2.RESTRIÇÕES DO PROBLEMA

Para que uma solução proveniente do algoritmo genético seja considerada como uma significativa melhoria para o ambiente fabril, é essencial que não apenas otimize o desempenho

na linha de produção, mas também que seja uma solução viável para implementação efetiva no chão de fábrica. Para alcançar esse objetivo, é imperativo incorporar todas as restrições físicas pertinentes ao algoritmo de busca, de modo a buscar a solução mais adequada à realidade operacional.

No contexto do problema específico de posicionamento de sensores ao longo das esteiras da linha de produção, as restrições podem ter origens diversas, como a presença de mesas de ar, que são esteiras capaz de criar um fluxo de ar que passa por furos na superfície da mesa, permitindo que as latas levitem sobre o ar, reduzindo o atrito e facilitando o transporte. A existência de motores no espaço físico, que impedem a instalação de sensores no mesmo local. E por também a necessidade de manter uma distância mínima em relação à entrada/saída da máquina. A abordagem para incorporar essas restrições ao algoritmo genético implica a definição de limites para a localização que cada sensor pode ocupar em uma esteira. Isso assegura que locais indisponíveis sejam adequadamente alocados e que mais de um sensor não ocupe o mesmo espaço, garantindo assim a conformidade com as limitações físicas existentes.

Além disso, é fundamental considerar as configurações específicas das esteiras no modelo desta linha de produção. A análise minuciosa dessas configurações permitirá uma adaptação precisa do algoritmo genético às características particulares do ambiente fabril, contribuindo assim para a eficácia da solução proposta. Essa abordagem mais detalhada e específica visa assegurar não apenas a otimização do desempenho, mas também a viabilidade prática da implementação da solução no contexto operacional da fábrica. Por se tratar de uma empresa privada as informações de tamanho de esteiras e quantidade de sensores são sensíveis à confidencialidade da fábrica e, portanto, não serão revelados

Em conjunto com um operador encarregado do controle da linha de produção, procedemos à alocação das restrições específicas de cada esteira no algoritmo genético. A integração dessas restrições é fundamental para garantir a conformidade e a eficácia do algoritmo em face das particularidades de cada esteira. As tabelas 1 a 18 detalham as limitações de posição de cada esteira, com exceção das esteiras 9, 15,16 e 17, que devido a presença de mesa de ar não pode sofrer alteração na posição de seus sensores., a numeração de cada esteira está seguindo uma ordem genérica e não representa a sequência real de esteira da linha de produção.

Tabela 1 – Configurações de sensores da esteira 1

| Esteira 1 | | |
|------------------|---------------------------|---------------------------|
| Sensor | Posição mínima (m) | Posição máxima (m) |
| 1 | 0 | 5 |
| 2 | 6,7 | 10 |
| 3 | 11 | 14 |
| 4 | 15 | 24 |
| 5 | 25 | 29,5 |

Fonte: O Autor(2024).

Tabela 2 – Configurações de sensores da esteira 2

| Esteira 2 | | |
|------------------|---------------------------|---------------------------|
| Sensor | Posição mínima (m) | Posição máxima (m) |
| 1 | 33 | 39 |

Fonte: O Autor(2024).

Tabela 3 – Configurações de sensores da esteira 3

| Esteira 3 | | |
|------------------|---------------------------|---------------------------|
| Sensor | Posição mínima (m) | Posição máxima (m) |
| 1 | 1 | 3 |
| 2 | 3,1 | 4 |
| 3 | 4,1 | 5 |
| 4 | 5 | 6,5 |

Fonte: O Autor(2024).

Tabela 4 – Configurações de sensores da esteira 4

| Esteira 4 | | |
|------------------|--|--|
|------------------|--|--|

| Sensor | Posição mínima (m) | Posição máxima (m) |
|--------|--------------------|--------------------|
| 1 | 1 | 5 |

Fonte: O Autor(2024).

Tabela 5 – Configurações de sensores da esteira 5

| Esteira 5 | | |
|-----------|--------------------|--------------------|
| Sensor | Posição mínima (m) | Posição máxima (m) |
| 1 | 8 | 13 |
| 2 | 14 | 18 |
| 3 | 19 | 20.5 |
| 4 | 29 | 32,5 |

Fonte: O Autor(2024).

Tabela 6 – Configurações de sensores da esteira 6

| Esteira 6 | | |
|-----------|--------------------|--------------------|
| Sensor | Posição mínima (m) | Posição máxima (m) |
| 1 | 7 | 10 |

Fonte: O Autor(2024).

Tabela 7 – Configurações de sensores da esteira 7

| Esteira 7 | | |
|-----------|--------------------|--------------------|
| Sensor | Posição mínima (m) | Posição máxima (m) |
| 1 | 1 | 4 |
| 2 | 5 | 7 |
| 3 | 8 | 10 |
| 4 | 11 | 12 |

Fonte: O Autor(2024).

Tabela 8 – Configurações de sensores da esteira 8

| Esteira 8 | | |
|-----------|--------------------|--------------------|
| Sensor | Posição mínima (m) | Posição máxima (m) |
| 1 | 1 | 3,5 |
| 2 | 5 | 9 |
| 3 | 12 | 15 |
| 4 | 16 | 19 |

Fonte: O Autor(2024).

Tabela 9 – Configurações de sensores da esteira 9

| Esteira 9 | | |
|-----------|--------------------|--------------------|
| Sensor | Posição mínima (m) | Posição máxima (m) |
| 1 | - | - |
| 2 | - | - |

Fonte: O Autor(2024).

Tabela 10 – Configurações de sensores da esteira 10

| Esteira 10 | | |
|------------|--------------------|--------------------|
| Sensor | Posição mínima (m) | Posição máxima (m) |
| 1 | 4,5 | 7 |

Fonte: O Autor(2024).

Tabela 11 – Configurações de sensores da esteira 11

| Esteira 11 | | |
|------------|--------------------|--------------------|
| Sensor | Posição mínima (m) | Posição máxima (m) |
| 1 | 2 | 5,5 |

Fonte: O Autor(2024).

Tabela 12 – Configurações de sensores da esteira 12

| Esteira 12 | | |
|-------------------|---------------------------|---------------------------|
| Sensor | Posição mínima (m) | Posição máxima (m) |
| 1 | 4 | 9 |
| 2 | 10 | 17 |
| 3 | 18 | 25 |

Fonte: O Autor(2024).

Tabela 13 – Configurações de sensores da esteira 13

| Esteira 13 | | |
|-------------------|---------------------------|---------------------------|
| Sensor | Posição mínima (m) | Posição máxima (m) |
| 1 | 10 | 14 |
| 2 | 15 | 19 |
| 3 | 20 | 22 |
| 4 | 23 | 27 |

Fonte: O Autor(2024).

Tabela 14 – Configurações de sensores da esteira 14

| Esteira 14 | | |
|-------------------|---------------------------|---------------------------|
| Sensor | Posição mínima (m) | Posição máxima (m) |
| 1 | 1 | 4 |
| 2 | 5 | 7 |
| 3 | 8 | 11 |
| 4 | 15 | 18 |

Fonte: O Autor(2024).

Tabela 15 – Configurações de sensores da esteira 15

| Esteira 15 | | |
|-------------------|--|--|
|-------------------|--|--|

| Sensor | Posição mínima (m) | Posição máxima (m) |
|--------|--------------------|--------------------|
| 1 | - | - |
| 2 | - | - |

Fonte: O Autor(2024).

Tabela 16 – Configurações de sensores da esteira 16

| Esteira 16 | | |
|------------|--------------------|--------------------|
| Sensor | Posição mínima (m) | Posição máxima (m) |
| 1 | - | - |
| 2 | - | - |
| 3 | - | - |

Fonte: O Autor(2024).

Tabela 17 – Configurações de sensores da esteira 17

| Esteira 17 | | |
|------------|--------------------|--------------------|
| Sensor | Posição mínima (m) | Posição máxima (m) |
| 1 | - | - |
| 2 | - | - |
| 3 | - | - |

Fonte: O Autor(2024).

Tabela 18 – Configurações de sensores da esteira 18

| Esteira 18 | | |
|------------|--------------------|--------------------|
| Sensor | Posição mínima (m) | Posição máxima (m) |
| 1 | 1 | 6 |
| 2 | 7 | 12 |
| 3 | 13 | 18 |
| 4 | 19 | 24 |

| | | |
|---|------|----|
| 5 | 25 | 30 |
| 6 | 31,5 | 35 |
| 7 | 35,5 | 37 |

Fonte: O Autor(2024).

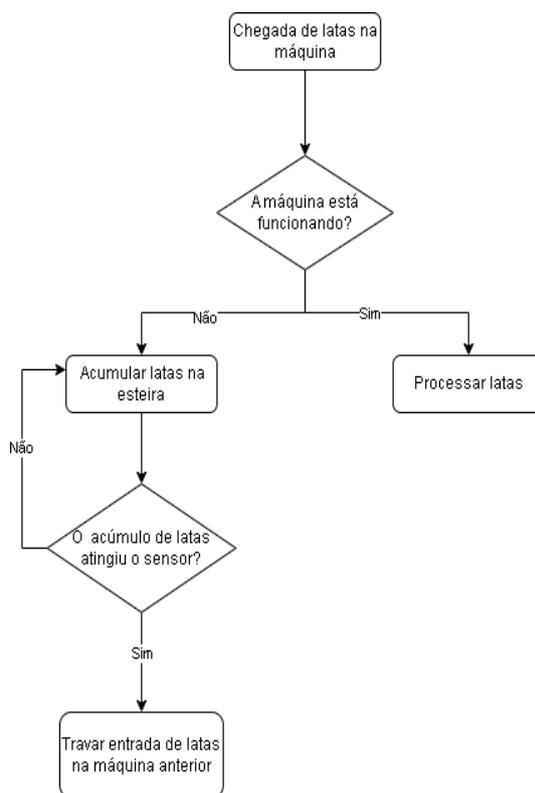
Ao segmentar as restrições de forma específica para cada esteira e sensor, buscamos otimizar a alocação dos sensores considerando as condições únicas de cada parte da linha de produção. Vale salientar que cada sensor poderá alterar sua posição entre o valor de sua posição máxima e mínima com intervalos de 0,01m, ou seja, o sensor entre 0 a 5m possui 500 posições diferentes, isso contribui para a criação de uma solução mais robusta e adaptada à complexidade das restrições físicas impostas em cada segmento da fábrica.

4.3.MODELAGEM DE FALHAS

No intuito de aprimorar a aderência do modelo de simulação e, conseqüentemente, a solução proporcionada pelo algoritmo genético, foi proposta uma correção na modelagem dos tempos de falha das máquinas dentro do contexto do modelo de simulação. A figura 23 mostra como a modelagem de falha pode impactar nas decisões que os sensores são responsáveis dentro do modelo. O melhor cenário possível acontece quando não existem falhas em todas as máquinas, o que resultaria em fluxo de latas sem interrupções. Quando ocorrem falhas aleatórias em alguma máquina, quanto maior sua duração, maior será o acúmulo de latas nas esteiras, o que implicará em trava de máquinas pelos sensores para evitar um acúmulo excessivo que geralmente ocasiona acidentes. A localização dos sensores irá determinar qual a proporção de esteira ocupada é necessária para emitir o sinal de trava/destrava de uma máquina.

Anteriormente, o método de representação dos tempos de falhas consistiam em uma adaptação que utilizava disponibilidade e tempos médios de reparo para determinar uma distribuição de probabilidade associada ao tempo entre falhas e ao tempo de reparo de cada máquina, um exemplo é demonstrado na figura 24. No entanto, considerando que cada máquina possui um comportamento de falhas distinto e complexo, torna-se indispensável realizar um estudo aprofundado sobre os modos de falha específicos de cada equipamento.

Figura 23 – Fluxograma de falhas sobre a decisão de sensores



Fonte: Elaboração própria

Por recomendação do próprio software utilizado, os tempos de falha e os intervalos entre falhas são modelados utilizando distribuições como Erlang, Exponencial Negativa ou Lognormal, pois segundo o guia do Plant simulation “são consideradas as mais adequadas para a modelagem de falhas de acordo com as características do sistema”. No entanto, por se tratar de um Processo não homogêneo de Poisson (PNHP), essas distribuições sugeridas podem não refletir com fidelidade o real comportamento das máquinas e, então, foi iniciado um esforço para aprimorar essa modelagem.

Figura 24 – Demonstração da modelagem de falhas atual

The screenshot shows a configuration dialog for a failure model in Plant Simulation. The fields are as follows:

- Name:** Failure
- Failed:**
- Active:**
- Start:** Const, 0
- Stop:** Const, 0
- Interval:** Negexp
- Duration:** Erlang
- Availability:**
- Availability:** [Progress bar] %
- MTTR:** [Progress bar]
- Failure relates to:** ProcessingTime

Buttons: OK, Cancel, Apply

Fonte: Captura de tela do software Plant Simulation

Para tal aprimoramento, busca-se uma abordagem mais precisa e personalizada, levando em consideração as características individuais de cada máquina, seus históricos de falhas e as condições específicas de operação. Isso envolve a aplicação de técnicas estatísticas avançadas e a realização de análises detalhadas para identificar padrões de falha e tendências em dados históricos. Ao compreender melhor os modos de falha e os padrões de comportamento das máquinas, será possível ajustar a modelagem dos tempos de falha de forma mais precisa, contribuindo assim para a eficácia do modelo de simulação utilizado no contexto industrial.

4.3.1. Independência entre falhas

Neste trabalho específico, supõe-se a independência entre as falhas. Essa hipótese implica que a ocorrência de uma falha em um componente não afeta a probabilidade de ocorrência de outras falhas em outros componentes (NANDA; GAYEN; CHOWDHURY, 2019). Em outras palavras, as falhas são tratadas como eventos independentes.

Ao assumir a independência de falhas, podem-se empregar métodos estatísticos e probabilísticos mais simples para estimar a confiabilidade de um sistema. Isso, geralmente, é feito através de modelos de confiabilidade baseados como a distribuição exponencial, a distribuição de Weibull, entre outras. Esses modelos são aplicados para prever a taxa de falha do sistema ao longo do tempo, levando em consideração a taxa de falha de cada componente individual, sem a necessidade de considerar interações complexas entre eles.

No entanto, é importante reconhecer que a suposição de independência de falhas nem sempre é realista em todos os contextos. Em sistemas altamente interconectados ou com componentes dependentes, as falhas em um componente podem desencadear falhas em outros componentes devido a efeitos de cascata ou dominó (LYU *et al.*, 2023). Nessas situações, a dependência entre falhas deve ser cuidadosamente considerada na análise de confiabilidade para evitar subestimar a probabilidade de falha do sistema.

Apesar dessa ressalva, a suposição de independência de falhas oferece várias vantagens significativas. Ela simplifica a modelagem e análise de sistemas, reduzindo a complexidade computacional e os requisitos de dados. Além disso, em muitos casos, essa suposição pode ser aceitável e razoavelmente precisa, especialmente quando aplicada a sistemas com componentes não relacionados ou operando em condições bem definidas e controladas. Assim, ao utilizar essa suposição de forma criteriosa e consciente, é possível obter insights valiosos sobre a confiabilidade e desempenho dos sistemas, facilitando a tomada de decisões em relação ao projeto, operação e manutenção.

4.3.2. Agrupamento de falhas

Após receber os dados de falhas da empresa em questão, foi identificada a necessidade de realizar uma análise detalhada e cuidadosa dos dados para extrair informações úteis e confiáveis. Com um total de 122 modos de falhas registrados, sendo 20 deles sem nome identificado, tornou-se evidente a importância de compreender a natureza e a distribuição dessas falhas para uma abordagem eficaz na gestão da manutenção e na melhoria do desempenho do sistema. As falhas sem nome representam um desafio significativo, sugerindo a possibilidade de modos de falhas não previstos ou desconhecidos pela empresa. Essa lacuna na identificação dos modos de falhas pode indicar uma falta de mapeamento completo do sistema, ressaltando a importância de revisar e atualizar os procedimentos de coleta e documentação de dados de falhas.

Para lidar com os modos de falhas registrados, foi adotada uma abordagem de clusterização, que agrupa modos de falhas semelhantes com base em suas características e padrões de ocorrência (DONG *et al.*, 2022). Essa estratégia permitiu identificar padrões subjacentes nos dados e reduzir a complexidade associada à grande variedade de modos de falhas listados. Ao agrupar modos de falhas semelhantes, mesmo que apresentem diferentes denominações, foi possível simplificar a análise e facilitar a interpretação dos resultados.

É importante ressaltar que não foi utilizado nenhum método específico para realizar a clusterização de falha. Este processo foi inteiramente realizado de forma empírica juntamente com a colaboração do operador mais experiente da linha de produção da empresa que desempenhou um papel fundamental nesse processo, fornecendo insights práticos sobre o funcionamento dos equipamentos e ajudando a validar as conclusões obtidas a partir da análise dos dados.

Cada máquina dentro do sistema apresenta um conjunto único de clusters de modos de falhas, refletindo de modo genérico as características de cada equipamento; a descrição de cada cluster por máquina é informado na tabela 19. Sendo assim, para cada cluster identificado, serão calculados uma distribuição que represente seu tempo de reparo e tempo entre falhas. Esses indicadores são essenciais para inserir o comportamento de falhas de cada máquina dentro do modelo de simulação, tornando-o cada vez mais aderentes ao real.

Tabela 19 – Clusterização de falhas

| Máquina | Cluster | Descrição |
|---------|---------|--------------------------|
| CUP | 1 | Falha desconhecida 1 |
| BM | 1 | Parada manual |
| | 2 | Lubrificação |
| | 3 | <i>Trimmer</i> |
| | 4 | Obstrução de latas |
| | 5 | Lata Com dimensão errada |
| | 6 | Falha elétrica |
| | 7 | Dispositivo de Segurança |
| | 8 | Ar |
| | 9 | Temperatura |

| Máquina | Cluster | Descrição |
|---------|---------|----------------------------------|
| | 10 | Motor |
| PT | 1 | Obstrução de latas |
| | 2 | Ar |
| | 3 | Ventilador |
| | 4 | Parada manual |
| | 5 | Verniz |
| | 6 | Forno |
| | 7 | Corrente |
| | 8 | Dispositivo de segurança |
| | 9 | Motor |
| | 10 | Conjunto de falhas desconhecidas |
| IS | 1 | Falha desconhecida 1 |
| | 2 | Falha desconhecida 2 |
| NCK | 1 | Obstrução de latas |
| | 2 | Pressão |
| | 3 | Ventilador |
| | 4 | Parada manual |
| | 5 | Pressco |
| | 6 | Dispositivo de Segurança |

| Máquina | Cluster | Descrição |
|---------|---------|----------------------------------|
| | 7 | Motor |
| | 8 | Conjunto de falhas desconhecidas |
| LT | 1 | Obstrução de latas |
| | 2 | Pressão |
| | 3 | Parada manual |
| | 4 | Pressco |
| | 5 | Dispositivo de Segurança |
| | 6 | Falha elétrica |
| PLT | 1 | Falha desconhecida 1 |
| WSH | 1 | Falha desconhecida 1 |
| | 2 | Falha desconhecida 2 |

Fonte: O Autor(2024).

4.3.3. Restrições do software

O processo de modelagem de falhas aleatórias requer uma análise cuidadosa das características das distribuições de probabilidade testadas, a fim de garantir uma modelagem precisa do processo em questão. No contexto do Plant Simulation, é importante considerar que o software oferece suporte a uma gama limitada de distribuições, destacadas na figura 25.

Dentre as distribuições suportadas, cinco se destacam para a modelagem de falhas: Exponencial negativa, Weibull, Lognormal, Triangular e Gamma. No entanto, apenas três dessas opções estão disponíveis na biblioteca do *Python* utilizada para realizar os testes de aderência: Gamma, Lognormal e Weibull. Portanto, este estudo restringirá a modelagem de falhas das máquinas neste contexto a essas distribuições. É importante ressaltar que essa

limitação pode resultar em erros, falta de aderência ou outras inadequações no processo modelado e pode não representar fielmente a realidade.

Figura 25 – Distribuições suportadas no Plant simulation

The screenshot shows a configuration window for a failure event in Plant Simulation. The event is named 'Obstrução'. It has options for 'Failed' and 'Active' (both unchecked). The start and stop times are set to 'Const' with a value of '0'. The interval and duration are also set to 'Const'. The availability is set to 'Normal' (unchecked). The failure relation is set to 'mulationTime'. The repair time (TR) is set to '1:00'. A dropdown menu for 'Failure relates' is open, listing various probability distributions: Normal, Lognorm, Negexp, Geom, Hypgeo, Erlang, Weibull, Triangle, Binomial, Poisson, Gamma, Beta, dEmp, cEmp, Emp, and Formula. The 'OK', 'Cancel', and 'Apply' buttons are at the bottom.

Fonte: Captura de tela do software Plant Simulation

4.3.4. Tempo de reparo

A distribuição de tempo correta para o tempo de reparo desempenha um papel crucial na precisão e na eficácia dos modelos de simulação, especialmente na representação precisa do comportamento estocástico da linha de produção. A escolha da distribuição adequada reflete a variabilidade do tempo de reparo das máquinas, permitindo uma modelagem mais precisa e realista das operações. Uma distribuição bem ajustada contribui para uma análise mais confiável dos tempos de inatividade e ajuda a identificar cenários potenciais que podem afetar o desempenho operacional.

Por outro lado, uma distribuição de tempo de reparo incorreta pode levar a estimativas imprecisas e distorcer os resultados da simulação. Por exemplo, se a distribuição subestimar a

variabilidade do tempo de reparo, o modelo pode prever tempos de inatividade mais curtos ou mais longos do que os observados na prática. Isso pode levar a decisões errôneas no controle da linha de produção, representando então um comportamento diferente do real, que por sua vez, irá interferir no resultado do algoritmo genético. Portanto, é essencial realizar uma análise cuidadosa e utilizar dados históricos ou estudos de tempos de reparo para selecionar a distribuição mais apropriada para o tempo de reparo no modelo de simulação.

Ao adequar a parte estocástica do modelo com um tempo de reparo através de distribuição de tempo correta, as organizações podem obter uma representação mais precisa e realista das operações industriais. Isso não apenas aumenta a confiabilidade das previsões e análises resultantes do modelo, mas também permite uma otimização mais eficaz nos estudos onde o modelo de simulação é utilizado.

Para obter as distribuições para os tempos de falhas, foram obtidos dados da operação de cada máquina entre 01/01/2023 até 07/02/2024. Com estes dados de falhas é possível determinar qual distribuição de probabilidade melhor modela os tempos de falha para cada máquina. Esta etapa foi realizada através de um código em *python*, que é disponibilizado no apêndice A, com ajuda da biblioteca *Reliability*, uma ferramenta criada para auxiliar nos estudos da engenharia da confiabilidade.

O comando *Fit_Everything* da biblioteca *reliability.Fitters* do *Python* é uma ferramenta valiosa utilizada na análise de confiabilidade para ajustar diversos modelos de distribuição de probabilidade aos dados de falha. Após a realização do ajuste dos modelos, esse comando oferece uma série de métricas que permitem avaliar a qualidade do ajuste realizado. Entre essas métricas, destacam-se o *Log-likelihood*, AIC, BIC e AD.

O *Log-likelihood* é uma medida fundamental que avalia o quão bem um modelo probabilístico se ajusta aos dados observados. Representa o logaritmo da função de verossimilhança, sendo que valores mais altos indicam um melhor ajuste do modelo aos dados (SO, 2003). O objetivo é maximizar o *Log-likelihood*, o que implica em minimizar a discrepância entre os dados observados e os valores previstos pelo modelo.

O AIC (*Akaike Information Criterion*) é outra métrica importante que considera tanto a qualidade do ajuste quanto a complexidade do modelo (A.-K. SEGHOUANE; AMARI, 2007). Penaliza modelos mais complexos para evitar o sobre ajuste, buscando minimizar seu valor. É

calculado a partir do *Log-likelihood*, com uma penalidade relacionada ao número de parâmetros do modelo.

O BIC (*Bayesian Information Criterion*) segue uma abordagem semelhante ao AIC, porém com uma penalidade mais rigorosa para modelos complexos (VRIEZE, 2012). Seu objetivo também é minimizar seu valor, indicando o melhor ajuste do modelo aos dados, levando em conta a complexidade do modelo.

A estatística de Anderson-Darling é um teste estatístico utilizado para verificar se um conjunto de dados segue uma determinada distribuição de probabilidade (LIEBSCHER, 2016). Quanto menor o valor da estatística de Anderson-Darling, melhor o ajuste do modelo aos dados. É calculada comparando as frequências observadas com as esperadas sob a distribuição de probabilidade ajustada. Essas métricas fornecem insights valiosos sobre a adequação dos modelos de distribuição de probabilidade aos dados de falha, facilitando a tomada de decisões na análise de confiabilidade e disponibilidade de sistemas. Vale ressaltar que os parâmetros das distribuições apresentadas estão ocultos devido à confidencialidade dos processos da empresa. A figura 26 aloca todos os clusters de cada máquina a uma distribuição, se houver, resultante do processo de modelagem. Em casos em que não foi suportado se refere à inadequação das distribuições testadas aos dados propostos.

Figura 26 – Alocação de cluster as distribuições (Tempo de reparo)

| Tempo de reparo | | | | | |
|-----------------|-------------------|---------|-------|-----------------|-------------|
| Cluster | Lognormal | Weibull | Gamma | Sem dados | Não suporta |
| CUP | 1 | | | | |
| BM11 | 1,2,3,4,5,7,10 | | | 6,8,9 | |
| BM12 | 1,2,3,4,5,6,7,8* | | | 9 e 10 | |
| BM13 | | | | Todos | |
| BM14 | 1,2,3,4,5,6,7 | | | 8,9,10 | |
| BM15 | 1,2,3,4,5,6,7,8* | | | 9 e 10 | |
| BM16 | 1,2,3,4,5,6,7,9 | 10 | | 8 e 10 | |
| BM17 | 1,2,3,4,5,6,7,8,9 | | | 10 | |
| BM18 | 1,2,3,4,5,6,7,8 | | | 9 e 10 | |
| WSH | 1 e 2 | | | | |
| PT1 | 1,2,4,5,6,7,8,10 | | 3* | 9 | |
| PT2 | 6 e 10 | | | 1,2,3,4,5,7,8,9 | |
| IS11 | 1 | | | | 2 |
| IS12 | 1 | | | | 2 |
| IS13 | 1 | | | | 2 |
| IS14 | 1 | | | | 2 |
| IS15 | 1 | | | | 2 |
| IS16 | 1 | | | | 2 |
| IS17 | 1 | | | | 2 |
| IS18 | 1 | | | | 2 |
| NCK | 1,2,4,5,6,7 | 3* | | | 8 |
| LT1 | 1,3,4,5,6 | 2* | | | |
| LT2 | 1,3,4,5,6 | | | 2 | |
| PLT | 1 | | | | 2 |

* Baixa quantidade de dados

Fonte: Elaboração própria

4.3.5. Tempo entre falhas

A distribuição para o tempo entre falhas é um aspecto crítico na modelagem estocástica de sistemas industriais, pois influencia diretamente a confiabilidade e disponibilidade dos equipamentos. A escolha da distribuição para o tempo entre falhas é essencial para garantir a precisão e a representatividade do modelo de simulação.

A utilização de uma distribuição inadequada para o tempo entre falhas pode levar a previsões imprecisas e distorções nos resultados da simulação. Se a distribuição subestimar ou superestimar a variabilidade do tempo entre falhas poderá levar a resultados equivocados sobre

os estudos de simulação. Portanto, é fundamental realizar uma análise detalhada dos dados históricos de falhas para selecionar a distribuição mais apropriada para o tempo entre falhas no modelo de simulação (LING *et al.*, 2023). Ao adequar a parte estocástica do modelo com o tempo entre falhas de distribuição de tempo correta, as organizações podem obter uma representação mais precisa e confiável da confiabilidade dos sistemas industriais.

Para obter as distribuições corretas utilizou-se o mesmo conjunto de dados disponibilizados pela empresa. Também foi utilizado o *python* em conjunto com a biblioteca *Reliability* para auxiliar neste procedimento; o código utilizado para este processo está no apêndice A deste trabalho. A alocação das distribuições modeladas para a modelagem do tempo entre falhas de cada cluster é disposta na figura 27.

Figura 27 – Alocação de cluster as distribuições para tempo entre falhas

| Tempo entre falhas | | | | | |
|--------------------|------------------|---------|--------|-----------------|-------------|
| Cluster | Lognormal | Weibull | Gamma | Sem dados | Não suporta |
| CUP | 1 | | | | |
| BM11 | 2,3,4,5 | 10 | 1 e 7 | 6,8,9 | |
| BM12 | 1,2,3,4,5,6,7,8* | | | 9 e 10 | |
| BM13 | | | | Todos | |
| BM14 | 3,4,5 | 1,2,6 | 7 | 8,9,10 | |
| BM15 | 1,3,4,5,6 | 2 | 7 e 8* | 9 e 10 | |
| BM16 | 1,3,4,5 | 2 e 10 | 6 e 7 | 8 e 9 | |
| BM17 | 2,3,4,5,8,9 | 1, 6*,7 | | 10 | |
| BM18 | 3,4,6,8 | 1,2,5,7 | | 9 e 10 | |
| WSH | 2 | 1 | | | |
| PT1 | 3*,4,5,6,7,8,10 | 1 e 2 | | 9 | |
| PT2 | 10 | | 6 | 1,2,3,4,5,7,8,9 | |
| IS11 | | 1 | | | 2 |
| IS12 | 1 | | | | 2 |
| IS13 | 1 | | | | 2 |
| IS14 | | 1 | | | 2 |
| IS15 | | 1 | | | 2 |
| IS16 | 1 | | | | 2 |
| IS17 | | 1 | | | 2 |
| IS18 | | 1 | | | 2 |
| NCK | 1,4,5 | 2 e 7 | 3* e 6 | | 8 |
| LT1 | 2*,4 | 1,3,5 | 6 | | |

Fonte: Elaboração própria

4.4.CONFIGURAÇÃO DE HIPERPARÂMETROS

A escolha dos hiper parâmetros utilizados no GA é uma etapa crucial para o desempenho do algoritmo de busca. Qualquer alteração no valor desses parâmetros (aumentando ou diminuindo) afeta o resultado do GA negativamente ou positivamente, escolher os parâmetros corretos é uma tarefa complexa (HASSANAT *et al.*, 2019).

De acordo com Hassanat *et al* (2019), devemos usar as operações *crossover* e mutação juntos para termos maior chance de alcançarmos melhor resultados do que usando um deles sozinho, além disso otimizar os parâmetros de busca também demonstrou aumentar a eficiência do algoritmo no tempo de busca pela solução ótima. O *crossover* é mais eficiente do que a mutação em grandes tamanho de geração, por outro lado, a mutação foi mais eficiente em pequenas quantidades de indivíduos.

O mesmo autor também classifica como população pequena aquelas que contêm menos de 100 indivíduos, população média aquelas que contêm entre 100 e 200 indivíduos e populações grandes aquelas que apresentam mais de 200 indivíduos. Serão testados os dois casos extremos para buscar por uma melhor solução. Para cada tipo de população há um método melhor para estabelecer os valores de probabilidade e delta por geração, os métodos sugeridos por Hassanat *et al* (2019) serão apresentados a seguir.

4.4.1.Dynamic ILM/DHC

Comprovado maior eficiência em pequenas populações, este método consiste em começar com probabilidade de mutação (MR) igual a 0, e probabilidade de crossover (CR) igual a 1, e com o passar das gerações inverter esses valores até a última geração, quando MR = 1 e CR = 0. Para realizar este processo, precisamos inserir o valor do delta por geração, que descobrimos ao dividir a probabilidade total pela quantidade de gerações (Gn) usadas no algoritmo genético.

$$\Delta(Cross) = - \frac{CR}{Gn} \quad (2)$$

$$\Delta(Mut) = + \frac{MR}{Gn} \quad (3)$$

4.4.2.Dynamic DHM/ILC

Comprovado ter mais eficiência em populações grandes, este método é o exato oposto ao método anterior, neste caso devemos começar sua primeira geração com MR =1 e CR = 0, e

com o passar das gerações esta probabilidade vão se invertendo, da mesma forma, porém em sentido contrário, sendo assim temos:

$$\Delta(Cross) = +\frac{CR}{Gn} \quad (4)$$

$$\Delta(Mut) = -\frac{MR}{Gn} \quad (5)$$

4.5.ANÁLISE DE RESULTADOS

Para a execução do algoritmo genético no Plant Simulation, empregou-se um computador com as seguintes especificações: *Intel Core i7-9700k 3.6GHz, 32gb de RAM, Nvidia GeForce RTX 2060, 240gb de SSD e 4tb de HD*. Diante das restrições inerentes ao problema, o Plant simulation disponibiliza a quantidade de posições que cada sensor poderá ocupar, estes valores são destacados na tabela 20.

Tabela 20 – Quantidade de posições possíveis de cada sensor

| Sensor | Quantidade de posições |
|-----------------------|------------------------|
| Esteira 1.sensorID(1) | 501 |
| Esteira 1.sensorID(2) | 331 |
| Esteira 1.sensorID(3) | 301 |
| Esteira 1.sensorID(4) | 900 |
| Esteira 1.sensorID(5) | 450 |
| Esteira 2.sensorID(1) | 601 |
| Esteira 3.sensorID(1) | 201 |
| Esteira 3.sensorID(2) | 91 |
| Esteira 3.sensorID(3) | 91 |
| Esteira 3.sensorID(4) | 141 |
| Esteira 4.sensorID(1) | 401 |

| | |
|------------------------|-----|
| Esteira 5.sensorID(1) | 501 |
| Esteira 5.sensorID(2) | 400 |
| Esteira 5.sensorID(3) | 150 |
| Esteira 5.sensorID(4) | 350 |
| Esteira 6.sensorID(1) | 301 |
| Esteira 7.sensorID(1) | 301 |
| Esteira 7.sensorID(2) | 201 |
| Esteira 7.sensorID(3) | 201 |
| Esteira 7.sensorID(4) | 101 |
| Esteira 8.sensorID(1) | 251 |
| Esteira 8.sensorID(2) | 401 |
| Esteira 8.sensorID(3) | 301 |
| Esteira 8.sensorID(4) | 300 |
| Esteira 10.sensorID(1) | 351 |
| Esteira 11.sensorID(1) | 501 |
| Esteira 12.sensorID(2) | 700 |
| Esteira 12.sensorID(3) | 700 |
| Esteira 12.sensorID(1) | 251 |
| Esteira 13.sensorID(1) | 401 |
| Esteira 13.sensorID(2) | 400 |
| Esteira 13.sensorID(3) | 200 |
| Esteira 13.sensorID(4) | 400 |

| | |
|------------------------|-----|
| Esteira 14.sensorID(1) | 301 |
| Esteira 14.sensorID(2) | 201 |
| Esteira 14.sensorID(4) | 301 |
| Esteira 14.sensorID(3) | 300 |
| Esteira 18.sensorID(1) | 501 |
| Esteira 18.sensorID(2) | 501 |
| Esteira 18.sensorID(3) | 500 |
| Esteira 18.sensorID(4) | 500 |
| Esteira 18.sensorID(5) | 500 |
| Esteira 18.sensorID(6) | 351 |
| Esteira 18.sensorID(7) | 151 |

Fonte: O Autor(2024).

Com isso possível delinear o espaço de soluções, que pode ser obtido utilizando a equação (1), obtendo aproximadamente $1,019^{110}$, cuja análise individual demandaria anos para conclusão.

A avaliação dos resultados será conduzida tanto para o algoritmo genético com populações reduzidas quanto ampliadas, com o intuito de discernir as disparidades nos resultados ao variar as configurações dos operadores genéticos. Ademais, será realizada uma comparação dos resultados obtidos após a integração da modelagem de falhas ao modelo, objetivando determinar se essa abordagem pode influenciar nas futuras otimizações do modelo de simulação.

4.5.1. Resultado sem falhas modeladas

No primeiro cenário, com uma população de 50 indivíduos e 10 gerações, foram gerados um total de 950 conjuntos de soluções a serem avaliados. A utilização dos operadores genéticos propostos pelo método *Dynamic ILM/DHC* resultou na redução do espaço de busca necessário

para 729 simulações, as quais foram concluídas em um tempo total de 2 horas, 13 minutos e 18 segundos. A tabela 21 apresenta as posições sugeridas pelo algoritmo para cada sensor:

Tabela 21 – Posição de sensores resultante 1

| Sensor | Posição (m) | Limite inferior (m) | Limite superior (m) |
|--------------------------|-------------|---------------------|---------------------|
| Esteira 1.sensorID(1) | 2,45 | 0 | 5 |
| Esteira 1.sensorID(2) | 7,84 | 6,7 | 10 |
| Esteira 1.sensorID(3) | 12,59 | 11 | 14 |
| Esteira 1.sensorID(4) | 20,32 | 15 | 24 |
| Esteira 1.sensorID(5) | 29,13 | 25 | 29,5 |
| Esteira 2.sensorID(1) | 34,14 | 33 | 39 |
| Esteira 3.sensorID(1) | 1,7 | 1 | 3 |
| Esteira 3.sensorID(2) | 3,73 | 3,1 | 4 |
| Esteira 3.sensorID(3) | 4,74 | 4,1 | 5 |
| Esteira 3.sensorID(4) | 6,06 | 5 | 6,5 |
| Esteira 4.sensorID(1) | 4,59 | 1 | 5 |

| | | | |
|--------------------------|-------|----|------|
| Esteira 5.sensorID(1) | 12,67 | 8 | 13 |
| Esteira 5.sensorID(2) | 14,66 | 14 | 18 |
| Esteira 5.sensorID(3) | 19,34 | 19 | 20,5 |
| Esteira 5.sensorID(4) | 31,14 | 29 | 32,5 |
| Esteira 6.sensorID(1) | 7,59 | 7 | 10 |
| Esteira 7.sensorID(1) | 3,6 | 1 | 4 |
| Esteira 7.sensorID(2) | 6,62 | 5 | 7 |
| Esteira 7.sensorID(3) | 9,96 | 8 | 10 |
| Esteira 7.sensorID(4) | 11,11 | 11 | 12 |
| Esteira 8.sensorID(1) | 1,95 | 1 | 3,5 |
| Esteira 8.sensorID(2) | 5,54 | 5 | 9 |
| Esteira 8.sensorID(3) | 12,47 | 12 | 15 |
| Esteira 8.sensorID(4) | 16,86 | 16 | 19 |

| | | | |
|---------------------------|-------|-----|-----|
| Esteira 10.sensorID(1) | 4,9 | 4,5 | 7 |
| Esteira 11.sensorID(1) | 4,9 | 2 | 5,5 |
| Esteira 12.sensorID(2) | 8,27 | 4 | 9 |
| Esteira 12.sensorID(3) | 15,69 | 10 | 17 |
| Esteira 12.sensorID(1) | 4,99 | 18 | 25 |
| Esteira 13.sensorID(1) | 13,89 | 10 | 14 |
| Esteira 13.sensorID(2) | 15,88 | 15 | 19 |
| Esteira 13.sensorID(3) | 20,19 | 20 | 22 |
| Esteira 13.sensorID(4) | 23,90 | 23 | 27 |
| Esteira 14.sensorID(1) | 1,77 | 1 | 4 |
| Esteira 14.sensorID(2) | 5,2 | 5 | 7 |
| Esteira 14.sensorID(4) | 10,63 | 8 | 11 |
| Esteira 14.sensorID(3) | 17,33 | 15 | 18 |

| | | | |
|---------------------------|-------|------|----|
| Esteira 18.sensorID(1) | 3,42 | 1 | 6 |
| Esteira 18.sensorID(2) | 8,13 | 7 | 12 |
| Esteira 18.sensorID(3) | 16,15 | 13 | 18 |
| Esteira 18.sensorID(4) | 21,36 | 19 | 24 |
| Esteira 18.sensorID(5) | 27,19 | 25 | 30 |
| Esteira 18.sensorID(6) | 34,63 | 31,5 | 35 |
| Esteira 18.sensorID(7) | 35,68 | 35,5 | 37 |

Fonte: O Autor(2024).

Esta solução encontrou um aumento de 0,57% na produção de latas do modelo de simulação. Tendo em vista a grande magnitude e capacidade de produção da indústria, um aumento como este somado ao fato de baixo valor necessário para ser investido é considerado muito bom.

No segundo cenário, com uma população de 200 indivíduos e 10 gerações, foram gerados um total de 3800 conjuntos de soluções a serem avaliados. A utilização dos operadores genéticos propostos pelo método *Dynamic DHM/ILC* não resultou na redução do espaço de busca, as quais foram concluídas em um tempo total de 12 horas, 39 minutos e 58 segundos. A tabela 22 apresenta as posições sugeridas pelo algoritmo para cada sensor:

Tabela 22 – Posição de sensores resultante 2

| Sensor | Posição (m) | Limite inferior (m) | Limite superior (m) |
|--------|-------------|---------------------|---------------------|
|--------|-------------|---------------------|---------------------|

| | | | |
|--------------------------|-------|-----|------|
| Esteira 1.sensorID(1) | 1,92 | 0 | 5 |
| Esteira 1.sensorID(2) | 6,77 | 6,7 | 10 |
| Esteira 1.sensorID(3) | 12,83 | 11 | 14 |
| Esteira 1.sensorID(4) | 17,91 | 15 | 24 |
| Esteira 1.sensorID(5) | 28,05 | 25 | 29,5 |
| Esteira 2.sensorID(1) | 33,84 | 33 | 39 |
| Esteira 3.sensorID(1) | 2,52 | 1 | 3 |
| Esteira 3.sensorID(2) | 3,23 | 3,1 | 4 |
| Esteira 3.sensorID(3) | 4,43 | 4,1 | 5 |
| Esteira 3.sensorID(4) | 5,27 | 5 | 6,5 |
| Esteira 4.sensorID(1) | 1,06 | 1 | 5 |
| Esteira 5.sensorID(1) | 8,14 | 8 | 13 |
| Esteira 5.sensorID(2) | 15,25 | 14 | 18 |

| | | | |
|---------------------------|-------|-----|------|
| Esteira 5.sensorID(3) | 19,56 | 19 | 20,5 |
| Esteira 5.sensorID(4) | 31,97 | 29 | 32,5 |
| Esteira 6.sensorID(1) | 9,73 | 7 | 10 |
| Esteira 7.sensorID(1) | 3,22 | 1 | 4 |
| Esteira 7.sensorID(2) | 5,52 | 5 | 7 |
| Esteira 7.sensorID(3) | 9,48 | 8 | 10 |
| Esteira 7.sensorID(4) | 11,77 | 11 | 12 |
| Esteira 8.sensorID(1) | 1,9 | 1 | 3,5 |
| Esteira 8.sensorID(2) | 5,51 | 5 | 9 |
| Esteira 8.sensorID(3) | 12,45 | 12 | 15 |
| Esteira 8.sensorID(4) | 17,26 | 16 | 19 |
| Esteira 10.sensorID(1) | 4,57 | 4,5 | 7 |
| Esteira 11.sensorID(1) | 6,29 | 2 | 5,5 |

| | | | |
|---------------------------|-------|----|----|
| Esteira 12.sensorID(2) | 14,65 | 4 | 9 |
| Esteira 12.sensorID(3) | 22,28 | 10 | 17 |
| Esteira 12.sensorID(1) | 5,78 | 18 | 25 |
| Esteira 13.sensorID(1) | 12,77 | 10 | 14 |
| Esteira 13.sensorID(2) | 15,38 | 15 | 19 |
| Esteira 13.sensorID(3) | 21,98 | 20 | 22 |
| Esteira 13.sensorID(4) | 23,39 | 23 | 27 |
| Esteira 14.sensorID(1) | 1,17 | 1 | 4 |
| Esteira 14.sensorID(2) | 5,27 | 5 | 7 |
| Esteira 14.sensorID(4) | 9,71 | 8 | 11 |
| Esteira 14.sensorID(3) | 16,07 | 15 | 18 |
| Esteira 18.sensorID(1) | 4,85 | 1 | 6 |
| Esteira 18.sensorID(2) | 9,2 | 7 | 12 |

| | | | |
|---------------------------|-------|------|----|
| Esteira 18.sensorID(3) | 15,28 | 13 | 18 |
| Esteira 18.sensorID(4) | 20,26 | 19 | 24 |
| Esteira 18.sensorID(5) | 27,24 | 25 | 30 |
| Esteira 18.sensorID(6) | 34,8 | 31,5 | 35 |
| Esteira 18.sensorID(7) | 36,12 | 35,5 | 37 |

Fonte: O Autor(2024).

Esta solução encontrou um aumento de 0,51% na produção de latas do modelo de simulação.

Ambos os métodos alcançaram soluções que prometem resultados semelhantes, com uma pequena vantagem na solução 1, embora apresentem conjuntos de posições de sensores distintos. A maior disparidade entre eles reside no tempo necessário para obter a solução desejada: ao empregar o método com um maior número de indivíduos, o processo do algoritmo foi aproximadamente 5,7 vezes mais demorado, alcançando um resultado comparativamente próximo ao obtido com a pequena quantidade de indivíduos.

4.5.2. Resultados com falhas modeladas

No primeiro cenário após a inclusão da modelagem de falhas, com uma população de 50 indivíduos e 10 gerações, foram gerados um total de 950 conjuntos de soluções a serem avaliados. A utilização dos operadores genéticos propostos pelo método *Dynamic ILM/DHC* resultou na redução do espaço de busca necessário para 831 simulações, as quais foram concluídas em um tempo total de 3 horas, 47 minutos e 39 segundos. A tabela 23 apresenta as posições sugeridas pelo algoritmo para cada sensor:

Tabela 23 – Posição de sensores resultante 3

| Sensor | Posição (m) | Limite inferior (m) | Limite superior (m) |
|--------------------------|-------------|---------------------|---------------------|
| Esteira 1.sensorID(1) | 4,05 | 0 | 5 |
| Esteira 1.sensorID(2) | 7,65 | 6,7 | 10 |
| Esteira 1.sensorID(3) | 12,17 | 11 | 14 |
| Esteira 1.sensorID(4) | 19,4 | 15 | 24 |
| Esteira 1.sensorID(5) | 27,13 | 25 | 29,5 |
| Esteira 2.sensorID(1) | 34,77 | 33 | 39 |
| Esteira 3.sensorID(1) | 2,85 | 1 | 3 |
| Esteira 3.sensorID(2) | 3,79 | 3,1 | 4 |
| Esteira 3.sensorID(3) | 4,23 | 4,1 | 5 |
| Esteira 3.sensorID(4) | 5,74 | 5 | 6,5 |
| Esteira 4.sensorID(1) | 3,34 | 1 | 5 |
| Esteira 5.sensorID(1) | 12,73 | 8 | 13 |

| | | | |
|---------------------------|-------|-----|------|
| Esteira 5.sensorID(2) | 17,86 | 14 | 18 |
| Esteira 5.sensorID(3) | 19,6 | 19 | 20,5 |
| Esteira 5.sensorID(4) | 32,1 | 29 | 32,5 |
| Esteira 6.sensorID(1) | 7,9 | 7 | 10 |
| Esteira 7.sensorID(1) | 3,6 | 1 | 4 |
| Esteira 7.sensorID(2) | 5,24 | 5 | 7 |
| Esteira 7.sensorID(3) | 9,82 | 8 | 10 |
| Esteira 7.sensorID(4) | 11,22 | 11 | 12 |
| Esteira 8.sensorID(1) | 3,15 | 1 | 3,5 |
| Esteira 8.sensorID(2) | 5,55 | 5 | 9 |
| Esteira 8.sensorID(3) | 12,04 | 12 | 15 |
| Esteira 8.sensorID(4) | 18,45 | 16 | 19 |
| Esteira 10.sensorID(1) | 2,42 | 4,5 | 7 |

| | | | |
|---------------------------|-------|----|-----|
| Esteira 11.sensorID(1) | 7,06 | 2 | 5,5 |
| Esteira 12.sensorID(2) | 12,63 | 4 | 9 |
| Esteira 12.sensorID(3) | 21,49 | 10 | 17 |
| Esteira 12.sensorID(1) | 5,94 | 18 | 25 |
| Esteira 13.sensorID(1) | 11,83 | 10 | 14 |
| Esteira 13.sensorID(2) | 15,12 | 15 | 19 |
| Esteira 13.sensorID(3) | 21,3 | 20 | 22 |
| Esteira 13.sensorID(4) | 24,18 | 23 | 27 |
| Esteira 14.sensorID(1) | 3,82 | 1 | 4 |
| Esteira 14.sensorID(2) | 5,81 | 5 | 7 |
| Esteira 14.sensorID(4) | 9,79 | 8 | 11 |
| Esteira 14.sensorID(3) | 16,81 | 15 | 18 |
| Esteira 18.sensorID(1) | 2,83 | 1 | 6 |

| | | | |
|---------------------------|-------|------|----|
| Esteira 18.sensorID(2) | 8,5 | 7 | 12 |
| Esteira 18.sensorID(3) | 13,12 | 13 | 18 |
| Esteira 18.sensorID(4) | 21,71 | 19 | 24 |
| Esteira 18.sensorID(5) | 25,91 | 25 | 30 |
| Esteira 18.sensorID(6) | 32,13 | 31,5 | 35 |
| Esteira 18.sensorID(7) | 35,94 | 35,5 | 37 |

Fonte: O Autor(2024).

Esta solução encontrou um aumento de 1,89% na produção de latas do modelo de simulação.

No segundo cenário após a inclusão da modelagem de falhas, com uma população de 200 indivíduos e 10 gerações, foram gerados um total de 3800 conjuntos de soluções a serem avaliados. A utilização dos operadores genéticos propostos pelo método *Dynamic DHM/ILC* novamente não resultou na redução do espaço de busca, ao todo foi necessário um total de 17 horas, 33 minutos e 37 segundos para obter esta solução. A tabela 24 apresenta as posições sugeridas pelo algoritmo para cada sensor:

Tabela 24 – Posição de sensores resultante 4

| Sensor | Posição (m) | Limite inferior (m) | Limite superior (m) |
|--------------------------|-------------|---------------------|---------------------|
| Esteira 1.sensorID(1) | 3,16 | 0 | 5 |

| | | | |
|--------------------------|-------|-----|------|
| Esteira 1.sensorID(2) | 8,81 | 6,7 | 10 |
| Esteira 1.sensorID(3) | 13,52 | 11 | 14 |
| Esteira 1.sensorID(4) | 23,67 | 15 | 24 |
| Esteira 1.sensorID(5) | 29,18 | 25 | 29,5 |
| Esteira 2.sensorID(1) | 33,03 | 33 | 39 |
| Esteira 3.sensorID(1) | 1,51 | 1 | 3 |
| Esteira 3.sensorID(2) | 3,15 | 3,1 | 4 |
| Esteira 3.sensorID(3) | 4,57 | 4,1 | 5 |
| Esteira 3.sensorID(4) | 6,11 | 5 | 6,5 |
| Esteira 4.sensorID(1) | 3,05 | 1 | 5 |
| Esteira 5.sensorID(1) | 12,84 | 8 | 13 |
| Esteira 5.sensorID(2) | 17,95 | 14 | 18 |
| Esteira 5.sensorID(3) | 20,04 | 19 | 20,5 |

| | | | |
|---------------------------|-------|-----|------|
| Esteira 5.sensorID(4) | 31,32 | 29 | 32,5 |
| Esteira 6.sensorID(1) | 7,03 | 7 | 10 |
| Esteira 7.sensorID(1) | 3,99 | 1 | 4 |
| Esteira 7.sensorID(2) | 5,24 | 5 | 7 |
| Esteira 7.sensorID(3) | 9,19 | 8 | 10 |
| Esteira 7.sensorID(4) | 11,03 | 11 | 12 |
| Esteira 8.sensorID(1) | 1,61 | 1 | 3,5 |
| Esteira 8.sensorID(2) | 7,01 | 5 | 9 |
| Esteira 8.sensorID(3) | 13,61 | 12 | 15 |
| Esteira 8.sensorID(4) | 17,94 | 16 | 19 |
| Esteira 10.sensorID(1) | 4,64 | 4,5 | 7 |
| Esteira 11.sensorID(1) | 4,24 | 2 | 5,5 |
| Esteira 12.sensorID(2) | 16,39 | 4 | 9 |

| | | | |
|---------------------------|-------|----|----|
| Esteira 12.sensorID(3) | 18,67 | 10 | 17 |
| Esteira 12.sensorID(1) | 4,89 | 18 | 25 |
| Esteira 13.sensorID(1) | 10,92 | 10 | 14 |
| Esteira 13.sensorID(2) | 16,11 | 15 | 19 |
| Esteira 13.sensorID(3) | 21,73 | 20 | 22 |
| Esteira 13.sensorID(4) | 23,61 | 23 | 27 |
| Esteira 14.sensorID(1) | 3,62 | 1 | 4 |
| Esteira 14.sensorID(2) | 5,57 | 5 | 7 |
| Esteira 14.sensorID(4) | 9,02 | 8 | 11 |
| Esteira 14.sensorID(3) | 15,26 | 15 | 18 |
| Esteira 18.sensorID(1) | 4,93 | 1 | 6 |
| Esteira 18.sensorID(2) | 10,4 | 7 | 12 |
| Esteira 18.sensorID(3) | 16,3 | 13 | 18 |

| | | | |
|---------------------------|-------|------|----|
| Esteira 18.sensorID(4) | 22,83 | 19 | 24 |
| Esteira 18.sensorID(5) | 26,95 | 25 | 30 |
| Esteira 18.sensorID(6) | 32,23 | 31,5 | 35 |
| Esteira 18.sensorID(7) | 36,1 | 35,5 | 37 |

Fonte: O Autor(2024).

Esta solução encontrou um aumento de 1,79% na produção de latas do modelo de simulação.

A análise dos resultados com falhas modeladas demonstrou um padrão semelhante aos resultados sem falhas modeladas. Foi observado um aumento significativo no tempo necessário para obter a solução, além da ausência de redução do espaço de busca ao utilizar os hiperparâmetros recomendados. Além disso, foi notada uma tendência em que as soluções com 50 ou 200 indivíduos apresentaram retornos comparativamente próximos entre si, o que reduz a preferência por ampliar o espaço de busca para esse tipo de problema.

É importante destacar que ambos os cenários com modelagem de falhas demandaram mais tempo de processamento, sugerindo uma maior exigência computacional e, conseqüentemente, diminuindo a agilidade do modelo. A melhoria de resultados entre os testes com e sem falhas modeladas pode ser atribuída à possibilidade de que nem todos os modos de falha tenham sido incluídos no modelo, levando a uma disponibilidade de máquinas maior que a real, isso aconteceu pela disponibilidade incompleta e pouco detalhadas dos dados de falha fornecidos pela empresa.

5. CONCLUSÃO

Este estudo evidenciou que a combinação de técnicas de otimização combinatória com softwares de SED é altamente benéfica para os tomadores de decisão. Isso se deve não apenas à significativa redução no tempo necessário para encontrar soluções para problemas complexos, mas também à disponibilidade de uma ferramenta de apoio à decisão para testar propostas de solução, evitando potenciais riscos associados aos testes diretos no chão de fábrica. Nesse contexto, o algoritmo genético foi capaz de diminuir o tempo para obter uma solução viável de anos para pouco mais de 2 horas, ressaltando a importância de selecionar as ferramentas adequadas de acordo com a natureza do problema.

Além disso, observou-se um aumento na produção de latas em todos os resultados obtidos por meio do modelo de simulação, sugerindo que as configurações de sensores atualmente utilizadas podem não estar otimizadas, impactando negativamente na eficiência da linha de produção.

No que diz respeito às configurações de hiper parâmetros do algoritmo genético, o método com menor quantidade de indivíduos por geração mostrou-se mais eficaz, tanto na redução do espaço de busca necessário para encontrar uma solução viável quanto nas métricas usadas para avaliar a qualidade da solução. Portanto, ele apresentou um melhor desempenho em menos tempo, indicando uma preferência do Dynamic ILM/DHC para esse tipo de problema.

Quanto à modelagem de falhas, também foram observados indícios de que sua inclusão no modelo altera as soluções propostas, tanto no conjunto de soluções quanto no tempo necessário para obtê-las. A obtenção de dados mais completos e precisos sobre os registros de falhas demonstra ser importante para aumentar a confiança nos resultados do software, contribuindo para uma eficiência maior no apoio às decisões diárias dos operadores, esta etapa fica como sugestão para estudos futuros.

5.1. LIMITAÇÕES

O estudo enfrentou diversas limitações significativas, dentre as quais se destaca a restrição imposta pelo software Plant Simulation quanto ao uso de distribuições de falhas. Esta restrição obrigou a utilização de distribuições que podem não modelar adequadamente o evento de falha, o que, ao ser extrapolado para uma grande quantidade de máquinas, inevitavelmente afeta os resultados do modelo. Tal cenário resulta na diminuição da assertividade e confiança dos resultados obtidos, pois os números finais serão impactados diretamente.

Além disso, outra limitação foi identificada nos dados fornecidos pela empresa. Embora houvesse uma quantidade considerável de informações disponíveis, observaram-se várias lacunas significativas. Por exemplo, máquinas específicas como BM13 e PT2 não possuíam registros de falha, o que compromete a integridade dos resultados. Além disso, uma parcela substancial dos modos de falha não estava devidamente mapeada, com quase metade das máquinas não tendo uma destinação correta para os modos de falha existentes. Essa falta de informações detalhadas compromete sua profundidade e precisão. A consideração de independência entre falhas também é um fator limitante para este trabalho, pois essa afirmação muda completamente a forma de como proceder com a modelagem de falhas, o que poderia levar a resultados bastantes diferentes do que foi mostrado neste trabalho.

REFERÊNCIAS

A.-K. SEGHOUANE; AMARI, S.-I. The AIC Criterion and Symmetrizing the Kullback–Leibler Divergence. **IEEE Transactions on Neural Networks**, v. 18, n. 1, p. 97–106, 1 jan. 2007.

ALAM, T. et al. Genetic Algorithm: Reviews, Implementations, and Applications. 3 ago. 2020.

ALBUS, M. et al. Towards scalability for resource reconfiguration in robotic assembly line balancing problems using a modified genetic algorithm. **Journal of Intelligent Manufacturing**, 12 Jan. 2024.

ALHIJAWI, B.; AWAJAN, A. **Genetic algorithms: theory, genetic operators, solutions, and applications. Evolutionary Intelligence** Springer Science and Business Media Deutschland GmbH, , 2023.

ALLAN SARMENTO VIEIRA; WILSON FADLO CURI. EVIDENCIAÇÃO DOS MODELOS DE SIMULAÇÃO E TÉCNICAS PARA O PLANEJAMENTO INTEGRADO DOS SISTEMAS HÍDRICOS. **Revista Gestão & Sustentabilidade Ambiental**, v. 5, n. 1, p. 233–233, 12 maio 2016.

ARABI, B. H. **Solving NP-complete problems using Genetic Algorithms**. Proceedings - 2016 UKSim-AMSS 18th International Conference on Computer Modelling and Simulation, UKSim 2016. **Anais...**Institute of Electrical and Electronics Engineers Inc., 22 dez. 2016.

AZAB, E. et al. **Employing Genetic Algorithm and Discrete Event Simulation for Flexible Job-Shop Scheduling Problem**. Proceedings - 2021 International Conference on Computational Science and Computational Intelligence, CSCI 2021. **Anais...**Institute of Electrical and Electronics Engineers Inc., 2021.

BANKS, J. **Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice**. Wiley, 1998.

BANKS, J. et al. **Discrete-Event System Simulation**. 4. ed. Prentice Hall, 2004.

BANKS, J. et al. **Discrete-Event System Simulation**. 5. ed. Prentice-Hall, 2010.

BODENHOFER, U. **Genetic algorithms: theory and applications**. , 2004.

BRIGHENTI, José Renato Nunes. **Simulação e otimização de uma linha de manufatura em fase de projeto**. Orientador: MONTEVECHI, José Arnaldo Barra. 2006. Dissertação (Mestrado). Curso de Engenharia de Produção, Universidade Federal de Itajubá, Minas Gerais, 2006. Disponível em:

https://repositorio.unifei.edu.br/jspui/bitstream/123456789/3242/1/Dissertação_200630278.pdf
f. Acesso em: 28/06/2023.

CASELLA, G.; BERGER, R. L. **Statistical inference**. Belmont, Ca: Brooks/Cole Cengage Learning, 2017.

CHAKRABORTY, D.; RATHORE, P. K.; ROY, T. **Optimal Actuators Locations in Smart Fibre Reinforced Polymer Structures Using Genetic Algorithm**. (W. Fan, Ed.)MECHANICAL AND AEROSPACE ENGINEERING, PTS 1-7. **Anais...**: Applied Mechanics and Materials.2012.

CLAESKENS, G.; HJORT, N. L. **Model Selection and Model Averaging**. 2008.

CRUZ. **Globalização, inovação e sistema internacional: o Brasil e as transformações da economia mundial**. v. 1, n. 2, p. 165–178, 2 dez. 2011.

DE JONG, K. A.; SPEARS, W. M. A formal analysis of the role of multi-point crossover in genetic algorithms. **Annals of Mathematics and Artificial Intelligence**, v. 5, n. 1, p. 1–26, 1992.

DONG, Y. et al. Clustering method with axiomatization to support failure mode and effect analysis. **IISE Transactions**, p. 1–15, 6 jun. 2022.

ELHOUCINE OUASSAM et al. Heuristic Methods: Application to Complex Systems. 19 maio 2021.

EYRING, A. et al. Analysis of a closed-loop digital twin using discrete event simulation. **International Journal of Advanced Manufacturing Technology**, v. 123, n. 1–2, p. 245–258, 1 nov. 2022.

FERRO, R. et al. An optimization tool for production planning: A case study in a textile industry. **Applied Sciences (Switzerland)**, v. 11, n. 18, 1 set. 2021.

FERRO, R.; CORDEIRO, G. A.; ORDOÑEZ, R. E. C. **Dynamic modeling of discrete event simulation**. ACM International Conference Proceeding Series. **Anais...** Association for Computing Machinery, 8 jan. 2018.

FILIPPINI, R. Operations management research: some reflections on evolution, models and empirical studies in OM. **International Journal of Operations & Production Management**, v. 17, n. 7, p. 655–670, 1 jan. 1997.

FU, M. C. Feature Article: Optimization for simulation: Theory vs. Practice. **INFORMS Journal on Computing**, v. 14, n. 3, p. 192–215, ago. 2002.

GEN, M.; CHENG, R. Genetic algorithms and engineering design, 1997. **John Wiley and Sons, New York**, 1997.

GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization, and Machine Learning**. New York: Addison-Wesley, 1989.

GRZNÁR, P. et al. The use of a genetic algorithm for sorting warehouse optimization. **Processes**, v. 9, n. 7, 1 jul. 2021.

- GUIMARÃES, E. R. S. et al. Análise de Desempenho de Modelos de Otimização com Simulação a Eventos Discretos. **Iberoamerican Journal of Industrial Engineering**, v. 7, n. 13, p. 18–43, 30 jun. 2015.
- HARRELL, C.; GHOSH, B. K.; BOWDEN, R. **Simulation Using ProModel**. McGraw-Hill/Higher Education, 2004.
- HARRISON, J. et al. Simulation Modeling in Organizational and Management Research. **Academy of Management Review**, v. 32, 1 out. 2007.
- HASSAN, M.; SADEK, A.; ATTIA, H. A Real-Time Deep Machine Learning Approach for Sudden Tool Failure Prediction and Prevention in Machining Processes. **Sensors**, v. 23, n. 8, p. 3894–3894, 11 abr. 2023.
- HASSANAT, A. et al. Choosing mutation and crossover ratios for genetic algorithms-a review with a new dynamic approach. **Information (Switzerland)**, v. 10, n. 12, 1 dez. 2019.
- HILLIER, F. S.; LIEBERMAN, G. J. **Introduction to operations research**. New York, Ny: McGraw-Hill, 2021..
- HOLLAND, J. H. **Genetic Algorithms**. v. 267, n. 1, p. 66–73, 1992.
- KATOCH, S.; CHAUHAN, S. S.; KUMAR, V. A review on genetic algorithms: past, present, and future. **Multimedia Tools and Applications**, v. 80, n. 5, p. 8091–8126, 1 fev. 2021.
- LAW, A. M.; KELTON, W. D.; KELTON, W. D. **Simulation modeling and analysis**. Mcgraw-hill New York, 2007. v. 3
- LEAL, V. J.; ANDRADE, P. C. DE R. Modelagem dos dados de falha de um caminhão fora de estrada. **ForScience**, v. 6, n. 3, 18 dez. 2018.
- LEE, J.; BAGHERI, B.; KAO, H. A. A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. **Manufacturing Letters**, v. 3, p. 18–23, 1 jan. 2015.
- LIEBSCHER, E. Approximation of distributions by using the Anderson Darling statistic. **Communications in Statistics - Theory and Methods**, v. 45, n. 22, p. 6732–6745, 25 Jan. 2016.

LING, C. et al. Reliability Prediction of the Distribution Network Based on Wavelet Neural Network with Quantum Particle Swarm Optimization Algorithm. **Electric Power Components and Systems**, v. 51, n. 4, p. 398–408, 3 fev. 2023.

LYU, M. et al. Ripple effect in China–Europe Railway transport network: Ripple failure risk propagation and influence. **Physica A: Statistical Mechanics and its Applications**, v. 620, p. 128739–128739, 1 jun. 2023.

MIRANDA, R. D. C.; MONTEVECHI, J. A. B.; DE PINHO, A. F. Development of an adaptive Genetic algorithm for simulation optimization. **Acta Scientiarum - Technology**, v. 37, n. 3, p. 321–328, 1 jul. 2015.

MITCHELL, M. **An Introduction to Genetic Algorithms**. Cambridge: MIT Press, 1996.

NANDA, A. K.; GAYEN, S.; CHOWDHURY, S. Errors Due to Departure from Independence in Exponential Series System. **Cornell University**, 28 jan. 2019.

OBITKO, M. Introduction to genetic algorithms. **Czech Technical University: Prague, Czech Republic**, 1998.

OLYMPIA, R.; STEFKA, F.; PAPRZYCKI, M. **Influence of the population size on the genetic algorithm performance in case of cultivation process modelling**. In: FED, In Proc. 2013 Federated Conference on Computer Science and Information Systems, 371–376, 2013.

PEHERSTORFER, B.; KRAMER, B.; WILLCOX, K. Combining multiple surrogate models to accelerate failure probability estimation with expensive high-fidelity models. **Journal of Computational Physics**, v. 341, p. 61–75, jul. 2017.

REGINA MITSUE AZUMA. Otimização multiobjetivo em problema de estoque e roteamento gerenciados pelo fornecedor. 22 mar. 2021.

RÉGIS ANDRÉ WUTTKE; MIGUEL AFONSO SELLITTO. Cálculo da disponibilidade e da posição na curva da banheira de uma válvula de processo petroquímico. v. 8, n. 4, 26 nov. 2008.

SELLITTO, M. A. Análise estratégica da manutenção de uma linha de fabricação metal-mecânica baseada em cálculos de confiabilidade de equipamentos. **Revista Gestão da Produção, Operações e Sistemas**, n. 2, p. 97–97, 1 jun. 2007.

SIDERSKA, J. APPLICATION OF TECNOMATIX PLANT SIMULATION FOR MODELING PRODUCTION AND LOGISTICS PROCESSES. **Business, Management and Education**, v. 14, n. 1, p. 64–73, 10 jun. 2016.

SO, B. S. Maximized log-likelihood updating and model selection. **Statistics & Probability Letters**, v. 64, n. 3, p. 293–303, set. 2003.

SOUBHIA, A. L.; SERPA, A. L. Discrete optimization for actuator and sensor positioning for vibration control using genetic algorithms. **JVC/Journal of Vibration and Control**, v. 24, n. 17, p. 4050–4064, 1 set. 2018.

TAO, F. et al. Digital Twin in Industry: State-of-the-Art. **IEEE Transactions on Industrial Informatics**, v. 15, n. 4, p. 2405–2415, 1 abr. 2019.

TORTORELLA, G. L.; FETTERMANN, D. Implementation of Industry 4.0 and lean production in Brazilian manufacturing companies. **International Journal of Production Research**, v. 56, n. 8, p. 2975–2987, 19 out. 2017.

MERVE UZUNER ŞAHİN; ORHAN DENGİZ; DENGİZ, B. Optimization of the redundancy allocation problem: Genetic algorithm and Monte Carlo simulation with discrete events. **Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi**, v. 39, n. 1, p. 535–548, 21 ago. 2023.

VILLAGOMEZ, L. E. et al. **Discrete Event Simulation as a Support in the Decision Making to Improve Product and Process in the Automotive Industry - A Fuel Pump Component Case Study**. (L. M. Camarinha-Matos, H. Afsarmanesh, D. Antonelli, Eds.) 20th Working Conference on Virtual Enterprises (PRO-VE). **Anais...: Collaborative Networks and Digital Transformation**. Turin, Italy: Springer International Publishing, set. 2019. Disponível em: <https://inria.hal.science/hal-02478787>

VRIEZE, S. I. Model selection and psychological theory: A discussion of the differences between the Akaike information criterion (AIC) and the Bayesian information criterion (BIC). **Psychological Methods**, v. 17, n. 2, p. 228–243, 2012.

WANG, F.; XU, G.; WANG, M. An Improved Genetic Algorithm for Constrained Optimization Problems. **IEEE Access**, v. 11, p. 10032–10044, 2023.

APÊNDICE A

```
import pandas as pd
from datetime import datetime
from reliability.Fitters import Fit_Everything

# Funções para filtrar dados e calcular intervalos
def filtrar_e_transformar_para_lista(df, coluna, filtro1, filtro2):
    tabela_filtrada = df[(df['cMaquina'].str.contains(filtro1)) &
(df['Cluster'] == filtro2)]
    return tabela_filtrada[coluna].tolist()

def calcular_intervalo(data_inicial, data_final):
    h1 = datetime.strptime(data_inicial, "%d/%m/%Y %H:%M:%S")
    h2 = datetime.strptime(data_final, "%d/%m/%Y %H:%M:%S")
    diferenca_em_segundos = (h2 - h1).total_seconds()
    diferenca_em_horas = diferenca_em_segundos / 3600
    return diferenca_em_horas

# Leitura do arquivo Excel
try:
    df = pd.read_excel(caminho_arquivo_excel)
except FileNotFoundError:
    print("Arquivo não encontrado.")
    exit()

# Cálculo do MTTR
Duracao_falhas = filtrar_e_transformar_para_lista(df, 'Duracao',
filtro_maquina, filtro_cluster)
FIT_MTTR = Fit_Everything(failures= Duracao_falhas,
show_histogram_plot=True, show_probability_plot=False,
show_PP_plot=False, show_best_distribution_probability_plot=True,
exclude = (['Weibull_3P', 'Loglogistic_3P',
'Gamma_3P', 'Lognormal_3P', 'Weibull_Mixture', 'Weibull_DS', 'Weibull_CR', '
Gumbel_2P', 'Loglogistic_2P', 'Exponential_2P']))

# Cálculo do MTBF
tab_filt = filtrar_e_transformar_para_lista(df, 'DATA_STOP',
filtro_maquina, filtro_cluster)
tab_filt_sorted = tab_filt.sort_values(by='DATA_STOP')
```

```
datainit = "01/01/2023 00:00:00"
temp_ent_falhas = []

for index, linha in tab_filt_sorted.iterrows():
    data_hora = linha['DATA_STOP'].strftime('%d/%m/%Y %H:%M:%S')
    intervalo = calcular_intervalo(datainit, data_hora)
    temp_ent_falhas.append(float(f'{intervalo:.2f}'))
    datainit = linha['DATA_START'].strftime('%d/%m/%Y %H:%M:%S')

FIT_MTBF = Fit_Everything(failures= temp_ent_falhas,
show_histogram_plot=True, show_probability_plot=False,
show_PP_plot=False, show_best_distribution_probability_plot=True,
exclude = (['Weibull_3P', 'Loglogistic_3P',
'Gamma_3P', 'Lognormal_3P', 'Weibull_Mixture', 'Weibull_DS', 'Weibull_CR', '
Gumbel_2P', 'Loglogistic_2P', 'Exponential_2P']))

# Exibição dos resultados
print(f"MTTR: {FIT_MTTR}")
print(f"MTBF: {FIT_MTBF}")
```