# Otimização dos tempos de produção nas máquinas de enfesto e corte dos tecidos

Mateus C. Silva*

February 18, 2024

## 1 Introdução

Esse estudo de caso é feito no setor têxtil, sendo apresentada uma situação da empresa Têxtil Eficiente S.A, cujo é otimizar os tempos de produção nas máquinas de enfesto e corte dos tecidos.

A indústria têxtil faz parte da indústria de transformação e é uma das principais no setor quanto entre todos os setores no Brasil, tendo um faturamento passando de R$ 161 bilhões em 2020 (FuturePrint, 2023). Sendo o Vale do Itajaí (Santa Catarina) um dos principais polos na área (Tecnotêxtil, 2020). Contudo, houve um recuo do que era esperado para 2023, chegando a uma queda de produção de 9,3% nas fábricas de confecção de artigos de vestuário e acessórios, sendo uma perspectiva de crescimento modesto para 2024, algo entre 0,3% e 0,9% (Globo, 2024).

Contudo pode-se considerar ainda um cenário incerto tendo algumas medidas que ajudam o setor como manutenção da desoneração da folha de pagamento, mas outras que dificultam a competividade internacional como isenção de imposto de compras até U$ 50 dólares (Globo, 2024).

Com isso é possível notar que a adoção de medidas que otimizem a produção são essenciais para lidar com o cenário e fortalecer esse seguimento, nacionalmente quanto com a concorrência internacional.

### 1.1 Problema

O processo descrito em alto nível como: "no processo de enfesto é feita a disposição do tecido sobre uma mesa com uma máquina de enfesto que recebe um rolo de tecido e espalha o tecido sobre a mesa fazendo camadas. Em seguida, uma outra máquina faz o corte conforme moldes preestabelecidos. Após o corte feito as sobras são descartadas e as peças cortadas são levadas para o estoque ou etapa de costura.".

A Figura 1 ilustra o caso de estudo em que existem duas mesas de tamanho 10, tendo uma máquina de enfesto em cada mesa e uma máquina de corte que é compartilhada entre elas, contudo a mudança entre mesas tem um custo de tempo fixo. Para o planejamento, é dado um conjunto de ordens de produção $OP$ que devem ser escalonadas de forma que uma vez que ela vai para a mesa para a máquina de enfesto, ela só saí quando tiver sido concluído o corte.

*Institute of Computing, Universidade Federal da Bahia, Salvador, BA 40170-115, Brazil (mateuscsilva.1@gmail.com).

Vale ressaltar que as $OPs$ correspondem a um conjunto heterogêneo, sendo assim elas podem diferir entre si quanto ao espaço que ocupa, tempo médio de enfesto e tempo médio de corte. As máquinas de enfesto também podem diferir quanto ao tempo de *setup* que precede a realização de cada $OP$ pela máquina. Já a máquina de corte tem um tempo de *setup* fixo.

O objetivo é finalizar todas as $OPs$ no menor tempo possível respeitando todas as restrições do problema.

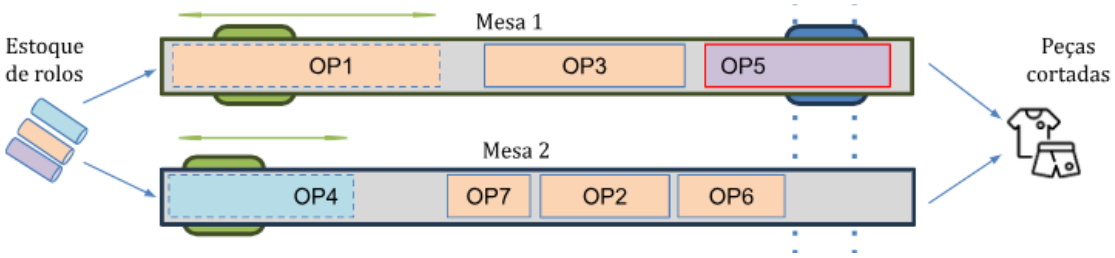


Figure 1: Exemplo de processo

### 1.2 Metodologia

O estudo foi dividido em três etapas. Na primeira e segunda etapa são analisadas versões simplificadas do problema e na terceira a versão completa. Para cada uma das etapas foi feito um modelo matemático para descrever o processo de otimização do problema de cada etapa. Além disso, para cada etapa foi desenvolvida uma solução metaheurística utilizando a metaheurística *biased random-key genetic algorithm* (BRKGA).

Os modelos foram implementados em Julia v1.6.7 utilizando a biblioteca JuMP e o *solver* Gurobi na versão 9.1.2. O BRKGA foi implementado em C++11 utilizando a API do BRKGA disponível em: https://github.com/rfrancotoso/brkgaAPI/. Os testes foram realizados numa máquina usando Windows 10, intel i5-8265U 1.6GHz com 8GB de RAM.

Para cada etapa foi fornecido um exemplo de entrada, sendo que não foi explicitado se os dados de exemplo podem ou representam alguma situação real, ou se são apenas ilustrativos para testes.

### 1.3 Organização

A seção 2 contém uma explicação sobre o funcionamento do BRKGA, assim como a justificativa da sua escolha e parâmetros utilizados. As demais se seções 3, 4 e 5 tem um resumo sobre cada etapa com a especificação do problema, resumo dos dados, informações sobre o modelo e sua execução, assim como para o BRKGA.

## 2 Biased random-key genetic algorithm

O algoritmo genético de chave aleatória enviesado (do inglês, *biased random-key genetic algorithm* - BRKGA) foi introduzido por Gonçalves and Resende (2011) como uma metaheurística de busca de uso geral para encontrar soluções de boa qualidade para problemas difíceis de otimização. O BRKGA simplifica algoritmos genéticos em geral, tornando tanto a representação quanto o mecanismo de intensificação-diversificação independentes do problema, como segue.

- Representação: As soluções no BRKGA são representadas como um vetor de chaves geradas aleatoriamente no intervalo [0, 1), seguindo o algoritmo genético de chaves aleatórias por Bean (1994). Tal vetor de chaves aleatórias define, ou codifica, uma solução única para o problema.
- Intensificação: no processo de cruzamento (*crossover*) do BRKGA, que produz parte da próxima geração de soluções, um dos pais é sempre uma solução da população elite (ou seja, uma com um alto valor de *fitness*), e a outra é uma solução não-elite (que não está na população de elite). Além disso, o O pai elite tem maior probabilidade de passar suas características (definidas pelas chaves) para os filhos.
- Diversificação: A cada geração, o BRKGA introduz novas soluções geradas aleatoriamente ções (ou seja, mutantes) na população. Isto evita a convergência prematura, permitindo que o algoritmo possa escapar de ótimos locais.

A principal tarefa é de decodificar a solução e calcular o seu *fitness*. Assim, o principal componente de uma implementação do BRKGA é o seu *decodificador*, ou seja, um algoritmo determinístico responsável por mapear o vetor de chaves aleatórias para uma possível solução do problema de otimização em questão. Dado um vetor de chaves aleatórias, o decodificador o mapeia para uma solução e calcula seu valor objetivo.

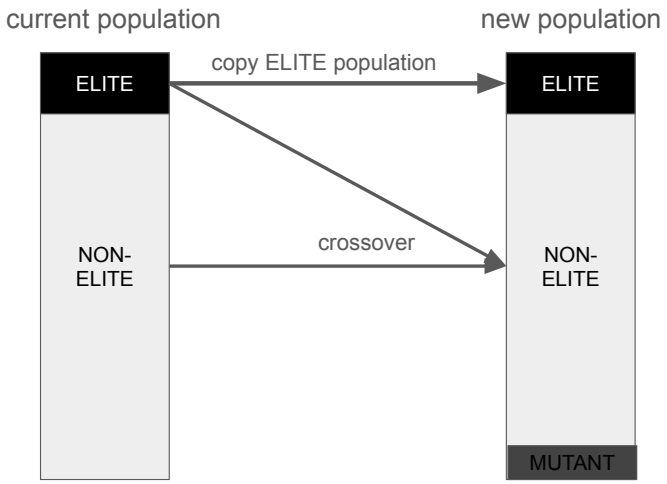


Figure 2: Evolução da população entre duas gerações no BRKGA

A figura 2 ilustra a etapa de evolução entre duas gerações. Além disso pode-se notar que o BRKGA tem uma estratégia elitista, sendo que o valor da melhor solução não decresce ao longo das gerações já que a população elite é copiada para a próxima geração. Em relação aos parâmetros, foi escolhido valores padrões seguindo o trabalho de Gonçalves and Resende (2011), sendo $p$ o tamanho da população, $p_e$ tamanho da população elite, $p_m$ tamanho da população mutante e $\rho_e$ a probabilidade de herança da solução elite. Por fim, ficaram $(p, p_e, p_m, \rho_e) = (\max(|n|, 100), 15\%, 10\%, 70\%)$.

O BRKGA apresenta diversas vantagens, sendo uma delas uma simplificação na implementação porque só precisa ser desenvolvido o *decoder*. Além disso, é possível realizar codificações e decodifi-

cações para o problema de forma que não exista restrição no espaço de busca, não exista soluções inviáveis durante o processo. Portanto sem necessidade de uso de penalidades que podem afetar a busca ou de correções para viabilizar uma solução que implicaria num custo adicional de tempo. Também, o BRKGA é uma metaheurística que encontra soluções muito boas e converge bem para soluções boas, em muitos casos ótimas Gonçalves and Resende (2011). Por fim, é fácil contornar no BRKGA problemas de convergência de ótimos locais ou guiar a busca através dos parâmetros ou caso necessário com a inserção de soluções inicias.

# 3 Etapa 1

A primeira versão do problema considera uma simplificação em que é dado um processo de enfesto de tecidos com duas máquinas $E1$ e $E2$, cada uma com um tempo de setup; além de um conjunto de ordens de produção ($OP$) de tecidos que devem ser realizados no menor tempo possível.

## 3.1 Resumo dos dados

2 Máquinas $E1$ e $E2$ com tempo de *setup* 2 e 3 respectivamente. São dadas 4 $OPs$ com tempos de enfesto 12,10,8,4.

## 3.2 Modelo

A descrição do modelo dessa etapa pode ser encontrado no arquivo **model_v0.pdf**. A figura 3 demonstra a execução do modelo, pode-se constatar a rápida convergência para a solução ótima. Pela figura 4 nota-se que foram encontradas duas soluções antes de encontra a solução ótima, levando menos de 1 segundo para provar a otimalidade como mostra o $gap = 0.0\%$. A solução mostra que a ordem selecionada foi OP1, OP2, OP3, OP4, sendo designadas para as máquinas 2, 1, 1, 2; respectivamente.

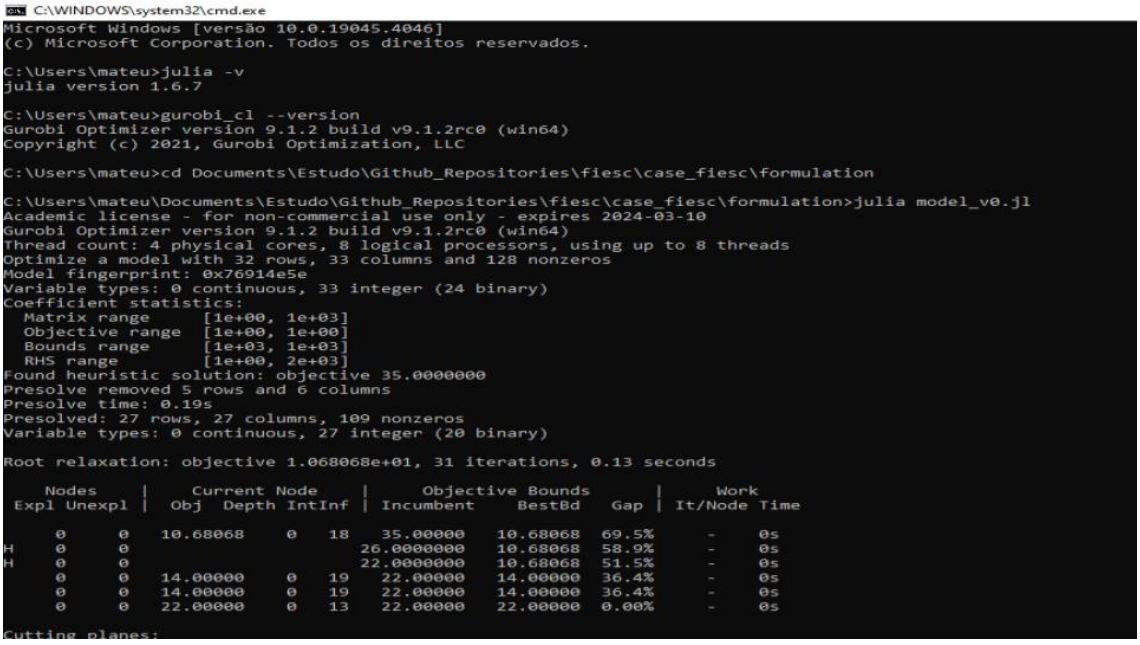


Figure 3: Execução do modelo v0

## 3.3 BRKGA

A figura 6 mostra que a melhor solução foi encontrada logo na primeira geração, além de ser a solução ótima que se sabe por ter sido provado pelo modelo. Além disso, as 10 melhores encontradas tiveram *fitness* iguais ao ótimo. O tempo de 10 segundo foi utilizado como critério de parada. A figura 5 mostra a codificação utilizada, as chaves na primeira metade (em cinza) representa a

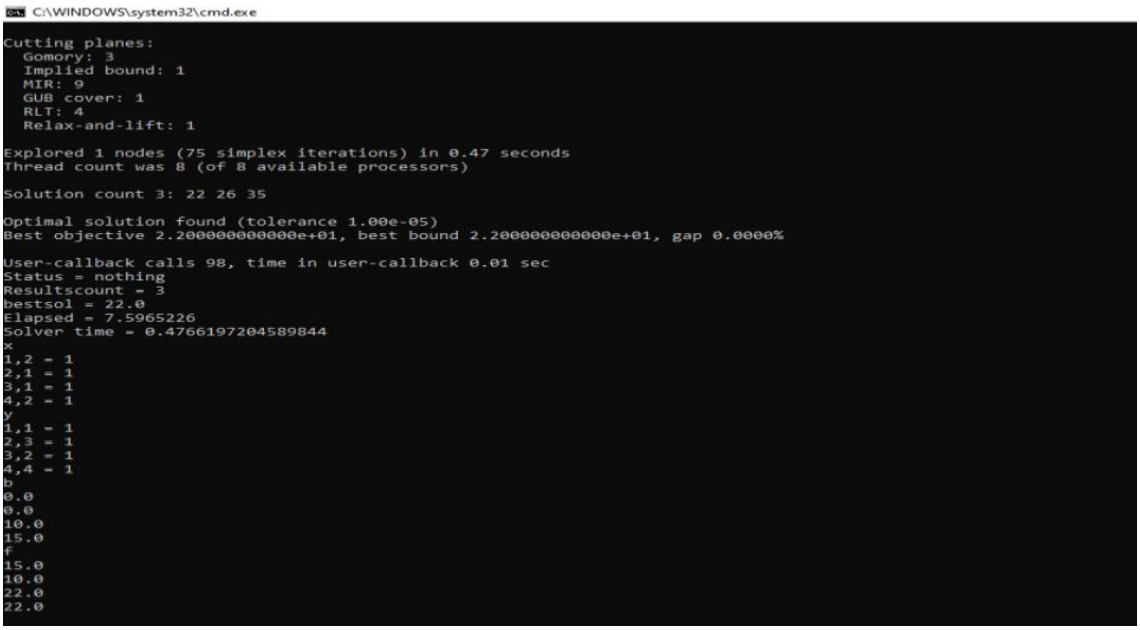


Figure 4: Resultado do modelo v0

prioridade da $OP$ aparecer primerio na ordem. A segunda metade indica para qual máquina de enfesto a $OP$ vai direcionada, se a chave for maior que 0.5 será para máquina 1, caso contrário para máquina 2.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| 0.23 | 0.38 | 0.11 | 0.68 | 0.48 | 0.97 | 0.62 | 0.03 |

Figure 5: Codificação da solucao

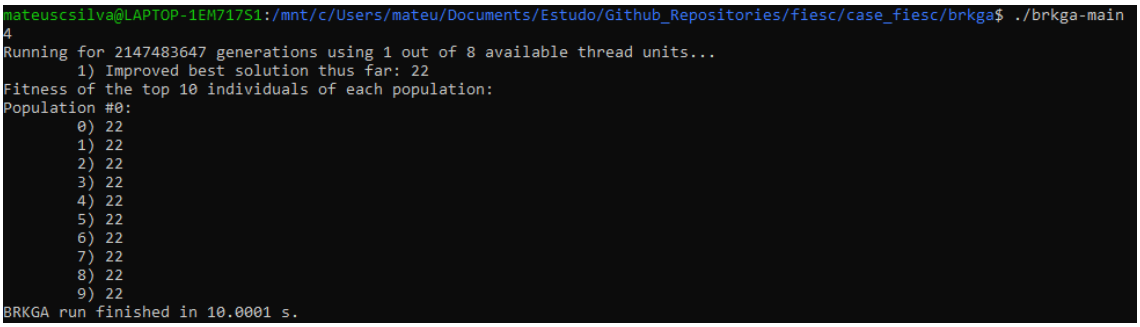


Figure 6: Resultado do BRKGA para Etapa 1

# 4 Etapa 2

A segunda versão do problema considera uma simplificação em que é dado um processo de enfesto de tecidos com uma máquina $E$ com tempo de *setup* $s_e$ e uma máquina de corte $C$ com tempo de *setup* $s_c$. O problema consiste em um conjunto de ordens de produção ($OP$) de tecidos que devem ser realizados em $E$ e depois cortadas por $C$ no menor tempo possível.

## 4.1 Resumo dos dados

1 Máquina $E$ com tempo de *setup* igual a 2. São dadas 4 $OPs$ com tempos de enfesto 18,14,12,6 e tempo de corte 4,4,6,6; respectivamente.

## 4.2 Modelo

A descrição do modelo dessa etapa pode ser encontrado no arquivo **model_v1.pdf**. A figura 7 demonstra a execução do modelo, pode-se constatar a rápida convergência para a solução ótima. A relaxação linear do problema para essa instância já levou ao resultado ótimo considerando um. Pela figura 8 nota-se que foram encontradas duas soluções, levando menos de 1 segundo para provar a otimalidade como mostra o $gap = 0.0\%$. A solução mostra que a ordem selecionada foi OP4, OP2, OP3, OP1, sendo os cortes iniciados logo após a conclusão do enfesto de cada $OP$.

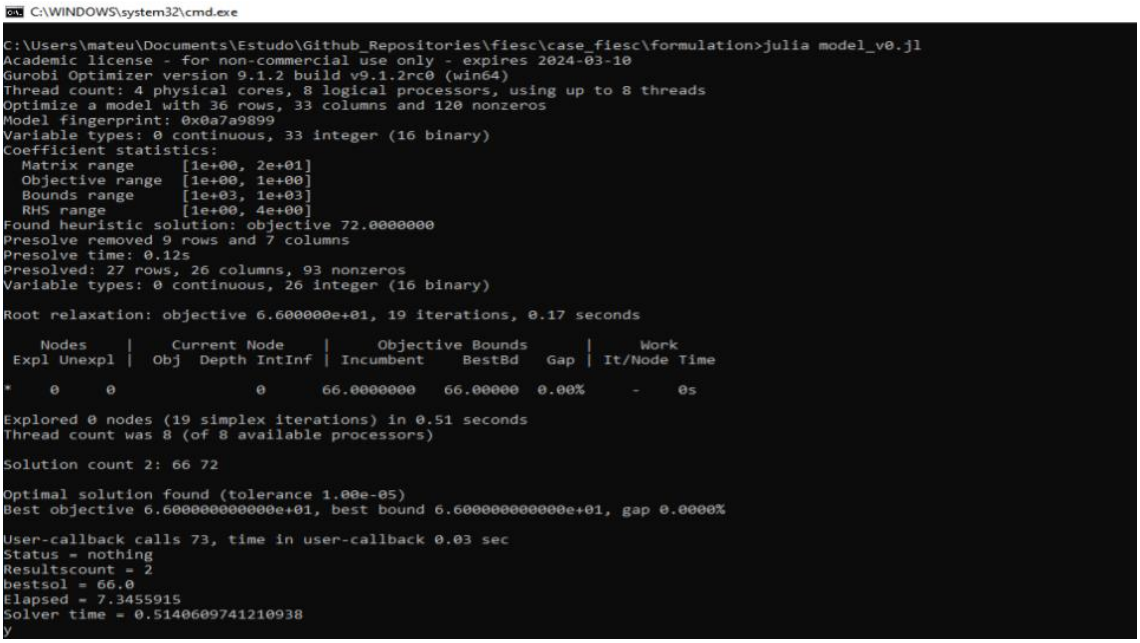


Figure 7: Execução do modelo v1

## 4.3 BRKGA

A figura 6 mostra que a melhor solução foi encontrada logo na primeira geração, além de ser a solução ótima que se sabe por ter sido provado pelo modelo. Além disso, as 10 melhores encontradas tiveram *fitness* iguais ao ótimo. O tempo de 10 segundo foi utilizado como critério de parada. A

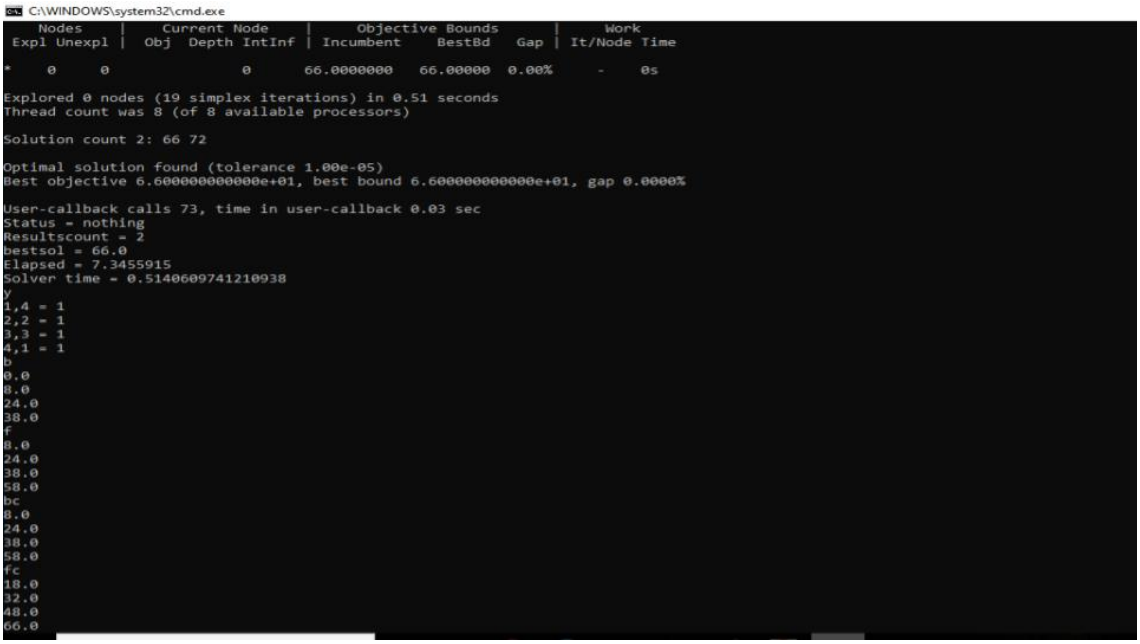


Figure 8: Resultado do modelo v1

figura 9 mostra a codificação utilizada, as chaves representam a prioridade da $OP$ aparecer primerio na ordem.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| 0.23 | 0.38 | 0.11 | 0.68 | 0.48 | 0.97 | 0.62 | 0.03 |

Figure 9: Codificação da solucao

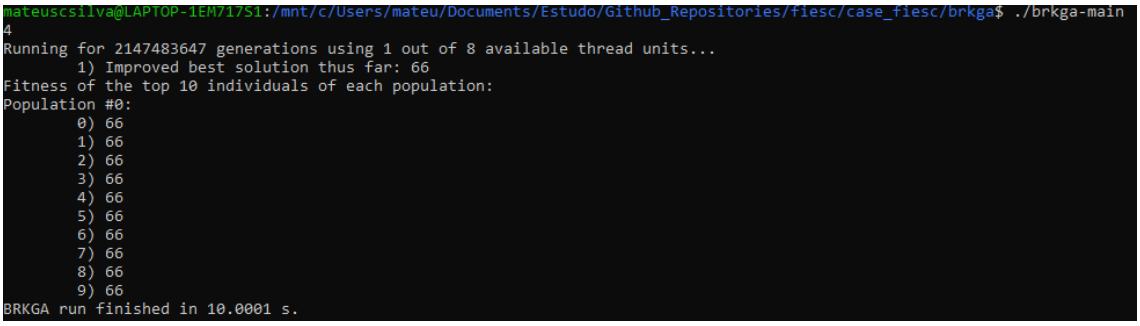


Figure 10: Resultado do BRKGA para Etapa 2

# 5 Etapa 3

A terceira versão do problema considera a versão completa descrita na Seção 1.1.

## 5.1 Resumo dos dados

2 máquinas $E$ com tempo de *setup* 2 e 3, respectivamente. Uma máquina de corte com tempo de *setup* de 4 e tempo de troca de mesa de 5. São dadas 9 $OPs$ com tempos de enfesto 18,14,12,6,36,9,9,12,12; tempo de corte 4,4,6,6,12,6,2,2,4; espaço ocupado 3,3,4,4,8,6,2,2,4 respectivamente.

## 5.2 Modelo

A descrição do modelo dessa etapa pode ser encontrado no arquivo **model_v2.pdf**. Foi descrito um modelo de programação lógica por restrições que engloba todas as restrições do problema, exceto a relacionada ao tempo de troca de mesa da máquina de corte. O modelo não foi completamente implementado devido a problemas ao utilizar o pacote *ConditionalJuMP* para implementar as restrições lógicas. Devido ao tempo não foi possível refazer usando outra linguagem e *solver*, como C++ e Cplex que eu sei que tem suporte para esse tipo de aplicação. Mais a baixo será descrito uma ideia de abordagem diferente para o problema que poderá ser implementada futuramente utilizando apenas programação inteira mista.

### 5.2.1 Abordagem futura alternativa

A ideia tem como base o trabalho desenvolvido por Barbosa et al. (2023). Abordar como um problema de roteamento e escalonamento de veículos capacitados. Dois veículos capacitados (seriam as mesas) que pegam os pedidos (OP) de um depósito. Cada veículo segue uma rota própria (máquina de enfesto). O veículo se desloca para um local de embarque, pois o local de entrega só é alcançável por uma barca (máquina de corte). A balsa leva um tempo para se deslocar para o ponto de partida de cada rota, caso ela não se encontre na estação.

A figura 11 ilustra como seria a instância adapatada. O primeiro nó a esquerda seria o ponto de partida. Cada veículo segue somente uma rota (superior ou inferior). O tempo de deslocamento para o primeiro par de nós seria o tempo de *setup* da máquina de enfesto que é representada na rota. O tempo de deslocamento do veículo entre o primeiro e segundo par representam o tempo de enfesto da última $OP$ no veículo. O trecho em azul seria o veículo sendo transportado pela balsa que chega ao ponto de entrega e libera a $OP$, ou seja, a $OP$ que já passou pelo enfesto só sai da mesa depois que terminou o corte. Além disso, o trecho em azul obriga que ao veículo esperar a balsa chegar ao ponto de embarque, que é equivalente a esperar o tempo de troca de mesa da máquina de corte caso seja necessário. Todos os arcos de retorno levam tempo 0 para sem atravessados, dessa forma não há alteração quanto aos valores da solução.

Essa abordagem poderia ser toda desenvolvida com programação inteira mista, retirando as restrições lógicas. A versão original conta com muitas restrições *BigM*, uma relaxação linear muito ruim, e exige muitos recursos computacionais. Contudo esses pontos talvez possam ser contornados por ter uma quantidade fixa de mesas, máquina de enfesto, corte e capacidade fixa.

Outra possibilidade poderia ser estender e juntar os trabalhos de Yang et al. (2003) e Silva and Subramanian (2019). O primeiro não conta com setup, já o segundo não conta com restrições de capacidade; ambos não possuem algo como uma máquina de corte.

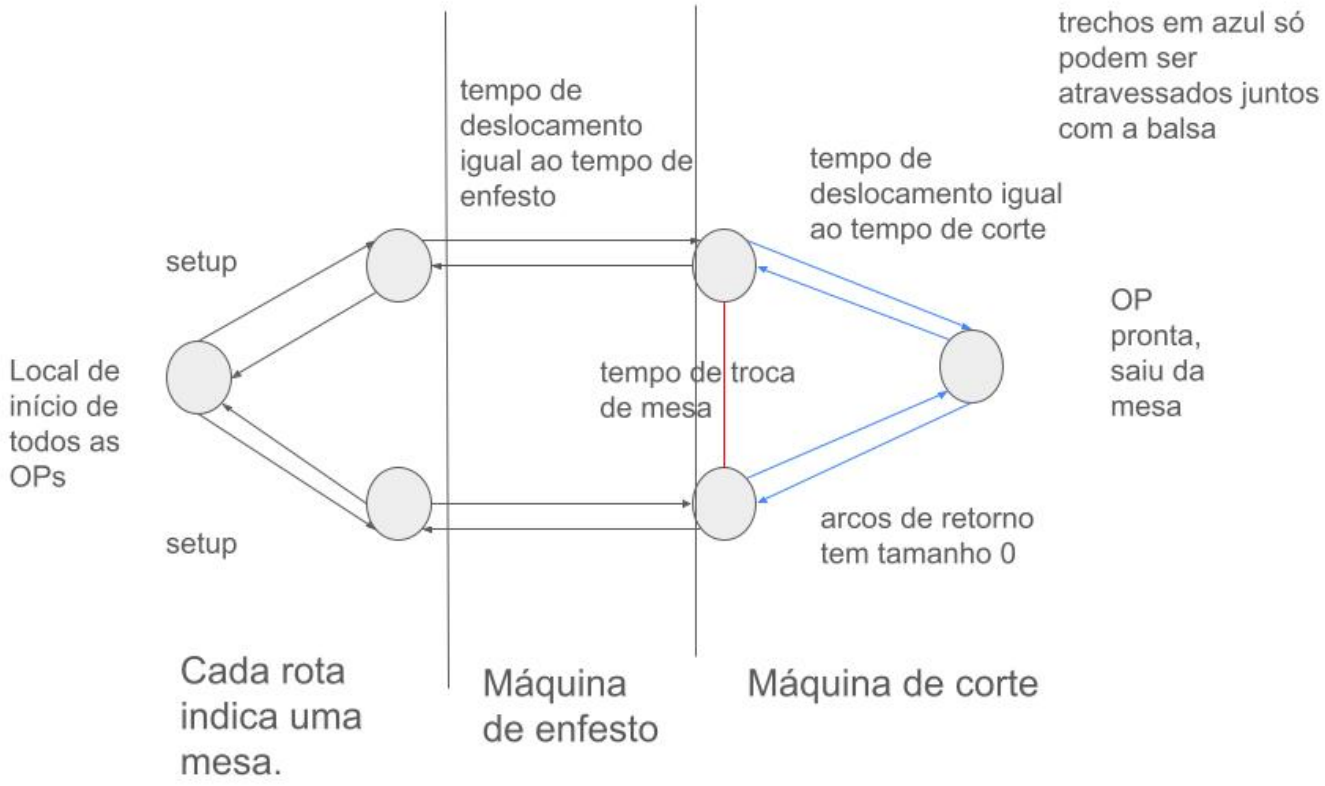


Figure 11: Ilustração de configuração de qualquer instância adaptada para o problema atual.

### 5.3 BRKGA

A figura 12 mostra a codificação da solução utilizada. O primeiro terço (em verde) do vetor a chave indica a ordem das $OPs$, o segundo terço (em azul) indica para qual máquina de enfesto a $OP$ vai ser direcionada, se a chave for maior que 0.5 será para máquina 1, caso contrário para máquina 2. A última parte (em laranja) indica em que mesa a máquina de corte estará a cada etapa, se a chave for maior que 0.5 será a primeira mesa, caso contrário irá para a segunda mesa.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|
| 0.23 | 0.38 | 0.11 | 0.68 | 0.48 | 0.97 | 0.62 | 0.03 | 0.58 |

Figure 12: Exemplo da codificação da solução no BRKGA

A figura 13 demonstra o resultado para o BRKGA para essa etapa. É possível notar que

o resultado é bem ruim, sendo na primeira geração indo até o tempo limite o que levanta duas possibilidades: o *decoder* proposto é muito ruim ou existe uma inconsistência que está deixando as *OPs* bloqueadas por muito mais tempo. Ainda é necessário desenvolver um validador e analisar esse caso, mas já se tem um código que permite fornecer um limite superior e fazer uma avaliação de pior caso.

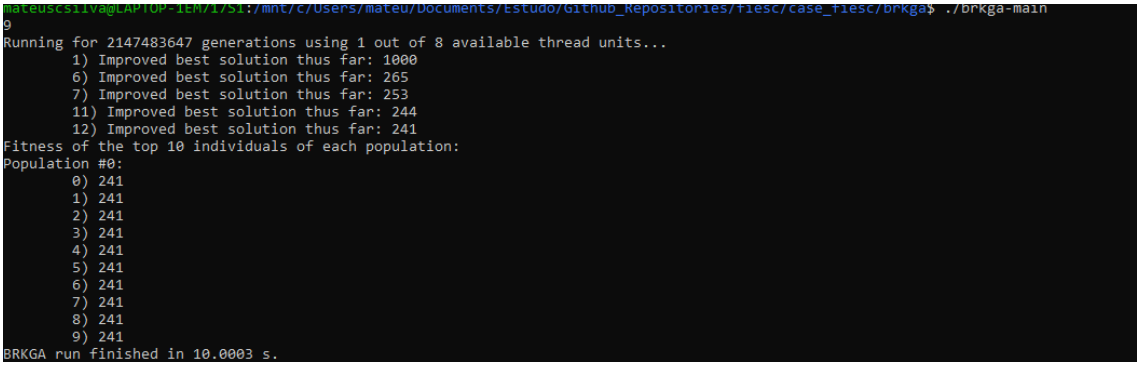


Figure 13: Resultado BRKGA para Etapa 3

## References

V. A. Barbosa, S. Tiwari, and R. Melo. O problema de coleta e entrega com janelas de tempo e escalonamento nas arestas. 2023.

J. C. Bean. Genetic algorithms and random keys for sequencing and optimization. *ORSA Journal on Computing*, 6(2):154–160, 1994.

R. FuturePrint. Saiba quais são os principais polos da indústria têxtil do brasil, 2023. Referência online em `https://digital.feirafutureprint.com.br/textil/industria-textil-os-impactos-e-o-futuro-deste-mercado-no-brasil`, último acesso em 18 de fevereiro, 2024.

Globo. Indústria têxtil vê reação tímida em 2024 e defende taxação de compras internacionais, 2024. Referência online em `https://revistapegn.globo.com/economia/noticia/2024/01/industria-textil-ve-reacao-timida-em-2024-e-defende-taxacao-de-compras-internacionais.ghtml`, último acesso em 18 de fevereiro, 2024.

J. F. Gonçalves and M. G. Resende. Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, 17(5):487–525, 2011.

J. M. P. Silva and A. Subramanian. Formulação matemática para o problema de escalonamento em máquinas paralelas idênticas com servidor único de setup. 2019.

Tecnotêxtil. Saiba quais são os principais polos da indústria têxtil do brasil, 2020. Referência online em `https://tecnotextilbrasil.com.br/`

saiba-quais-sao-os-principais-polos-da-industria-textil-do-brasil/, último acesso em 18 de fevereiro, 2024.

H. Yang, Y. Ye, and J. Zhang. An approximation algorithm for scheduling two parallel machines with capacity constraints. *Discrete Applied Mathematics*, 130(3):449–467, 2003.