

Otimização dos tempos de produção nas máquinas de enfiar e corte dos tecidos

Estudo de caso

Mateus Carvalho da Silva

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 enfiar e corte dos tecidos.

Introdução

- Faturamento no segmento passando R\$ 161 bilhões em 2020.
- Grande polo têxtil no Vale do Itajaí.
- Recuo de 9,3% nas fábricas de confecção de artigos de vestuário e acessórios em 2023.
- Desoneração da folha de pagamento
- Reforma tributária
- Isenção de imposto em compras internacionais até U\$ 50,00

Problema

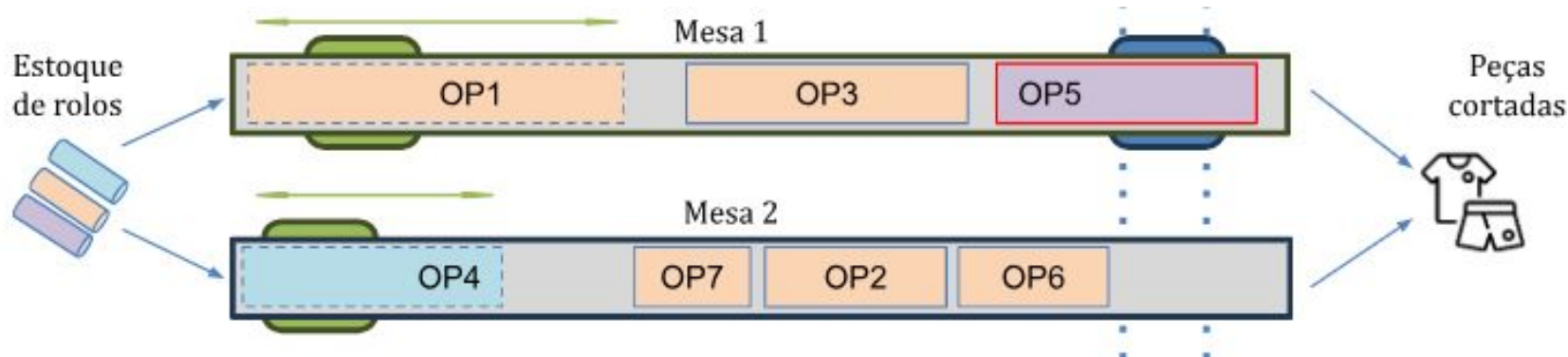


Ilustração do processo de enfiesto e corte. Exemplo fornecido pela FIESC.

Metodologia

- Estudo dividido em 3 etapas.
- 1 modelo matemático e 1 metaheurística para cada etapa.
- *Biased random-key genetic algorithm* (C++)
- Modelo desenvolvido em Julia usando JuMP.
- *Solver* Gurobi.

Metodologia

- Dados não coletados, fornecidos de exemplo.
- Ilustrativos ou possível exemplo real?
- Máquina para teste
 - Windows 10
 - intel i5-8625U, 1.6 GHz
 - 8Gb de RAM

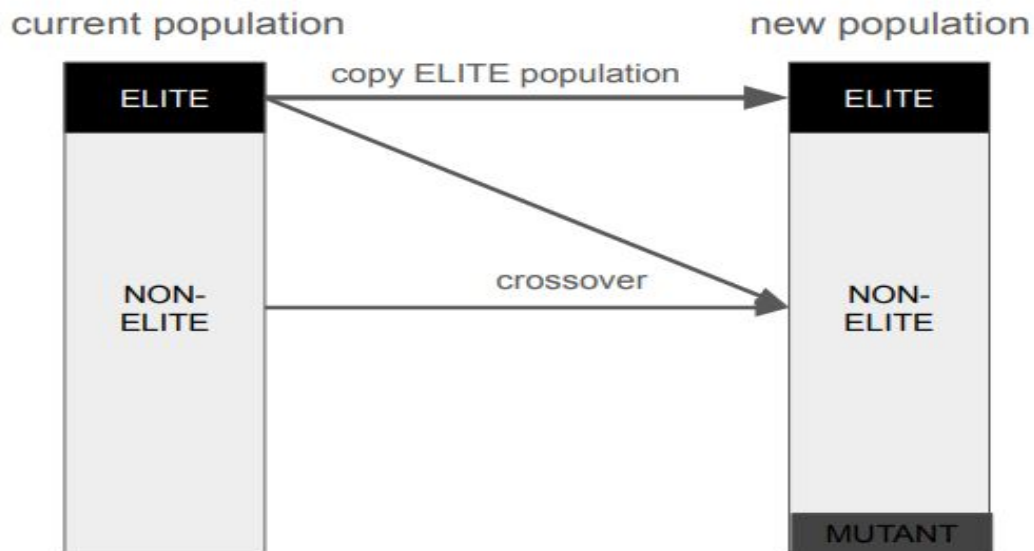
Biased random-key genetic algorithm (BRKGA)

Codificação

1	2	3	4	5	6
0.85	0.42	0.31	0.97	0.21	0.13

Biased random-key genetic algorithm (BRKGA)

Evolução entre duas gerações

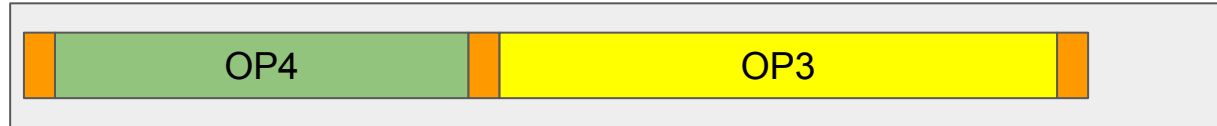
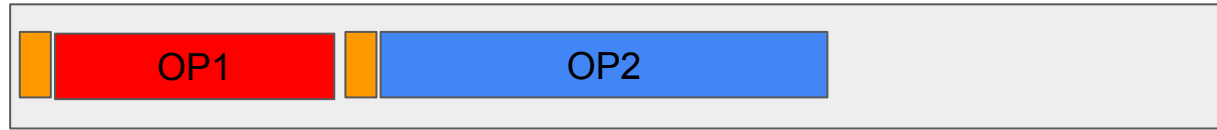


Biased random-key genetic algorithm (BRKGA)

Parâmetros

- p : tamanho da população = $\max(100, |OP|)$.
- p_e : tamanho da população elite = 15%.
- p_m : tamanho da população mutante = 10%.
- ρ_e : probabilidade de herança do pai elite = 70%.
- critério de parada: 10 segundos (exemplos pequenos)

Etapa 1



Resultado para o modelo

Microsoft Windows [versão 10.0.19045.4046]

(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\mateu>julia -v
julia version 1.6.7C:\Users\mateu>gurobi_cl --version
Gurobi Optimizer version 9.1.2 build v9.1.2rc0 (win64)
Copyright (c) 2021, Gurobi Optimization, LLC

C:\Users\mateu>cd Documents\Estudo\Github_Repositories\fiesc\case_fiesc\formulation

C:\Users\mateu\Documents\Estudo\Github_Repositories\fiesc\case_fiesc\formulation>julia model_v0.jl

Academic license - for non-commercial use only - expires 2024-03-10

Gurobi Optimizer version 9.1.2 build v9.1.2rc0 (win64)

Thread count: 4 physical cores, 8 logical processors, using up to 8 threads

Optimize a model with 32 rows, 33 columns and 128 nonzeros

Model fingerprint: 0x76914e5e

Variable types: 0 continuous, 33 integer (24 binary)

Coefficient statistics:

Matrix range [1e+00, 1e+03]

Objective range [1e+00, 1e+00]

Bounds range [1e+03, 1e+03]

RHS range [1e+00, 2e+03]

Found heuristic solution: objective 35.0000000

Presolve removed 5 rows and 6 columns

Presolve time: 0.19s

Presolved: 27 rows, 27 columns, 109 nonzeros

Variable types: 0 continuous, 27 integer (20 binary)

Root relaxation: objective 1.068068e+01, 31 iterations, 0.13 seconds

Nodes			Current Node			Objective Bounds			Work	
Expl	Unexpl		Obj	Depth	IntInf	Incumbent	BestBd	Gap	It/Node	Time
	0	0	10.68068	0	18	35.00000	10.68068	69.5%	-	0s
H	0	0				26.0000000	10.68068	58.9%	-	0s
H	0	0				22.0000000	10.68068	51.5%	-	0s
	0	0	14.00000	0	19	22.00000	14.00000	36.4%	-	0s
	0	0	14.00000	0	19	22.00000	14.00000	36.4%	-	0s
	0	0	22.00000	0	13	22.00000	22.00000	0.00%	-	0s

Cutting planes:

Cutting planes:

Gomory: 3
Implied bound: 1
MIR: 9
GUB cover: 1
RLT: 4
Relax-and-lift: 1

Explored 1 nodes (75 simplex iterations) in 0.47 seconds

Thread count was 8 (of 8 available processors)

Solution count 3: 22 26 35

Optimal solution found (tolerance 1.00e-05)

Best objective 2.200000000000e+01, best bound 2.200000000000e+01, gap 0.0000%

User-callback calls 98, time in user-callback 0.01 sec

Status = nothing

Resultscount = 3

bestsol = 22.0

Elapsed = 7.5965226

Solver time = 0.4766197204589844

x

1,2 = 1

2,1 = 1

3,1 = 1

4,2 = 1

y

1,1 = 1

2,3 = 1

3,2 = 1

4,4 = 1

b

0.0

0.0

10.0

15.0

f

15.0

10.0

22.0

22.0

BRKGA

Exemplo de codificação

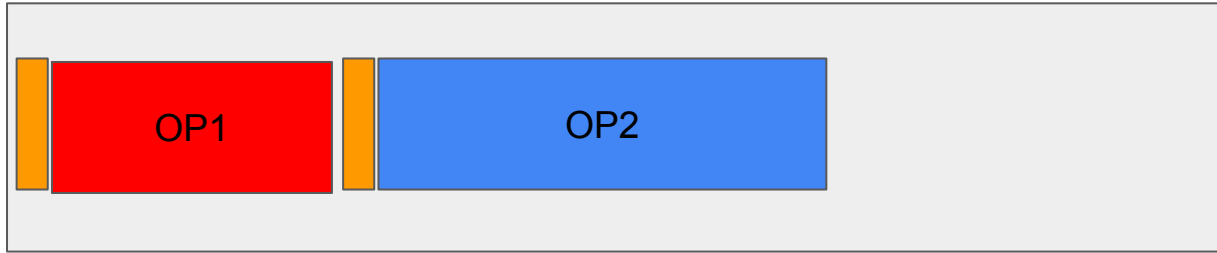
1	2	3	4	5	6	7	8
0.23	0.38	0.11	0.68	0.48	0.97	0.62	0.03

Resultado do BRKGA

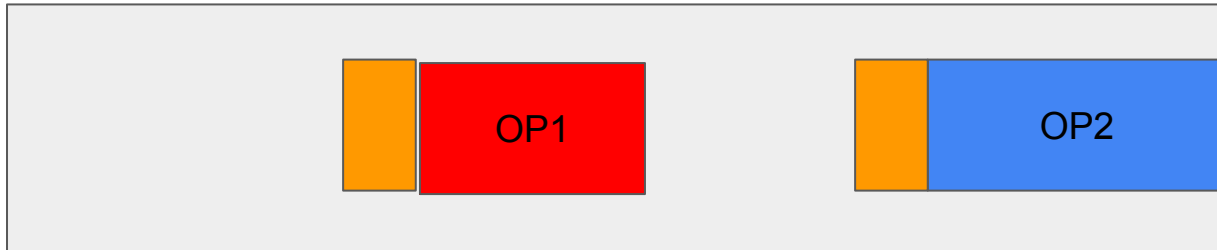
```
mateuscsilva@LAPTOP-1EM717S1:/mnt/c/Users/mateu/Documents/Estudo/Github_Repositories/fiesc/case_fiesc/brkga$ ./brkga-main
4
Running for 2147483647 generations using 1 out of 8 available thread units...
  1) Improved best solution thus far: 22
Fitness of the top 10 individuals of each population:
Population #0:
  0) 22
  1) 22
  2) 22
  3) 22
  4) 22
  5) 22
  6) 22
  7) 22
  8) 22
  9) 22
BRKGA run finished in 10.0001 s.
```


Etapa 2

Máquina de
enfesto



Máquina de
corte



Resultado para o modelo

C:\Users\mateu\Documents\Estudo\Github_Repositories\fiesc\case_fiesc\formulation>julia model_v0.jl

Academic license - for non-commercial use only - expires 2024-03-10

Gurobi Optimizer version 9.1.2 build v9.1.2rc0 (win64)

Thread count: 4 physical cores, 8 logical processors, using up to 8 threads

Optimize a model with 36 rows, 33 columns and 120 nonzeros

Model fingerprint: 0x0a7a9899

Variable types: 0 continuous, 33 integer (16 binary)

Coefficient statistics:

Matrix range [1e+00, 2e+01]

Objective range [1e+00, 1e+00]

Bounds range [1e+03, 1e+03]

RHS range [1e+00, 4e+00]

Found heuristic solution: objective 72.0000000

Presolve removed 9 rows and 7 columns

Presolve time: 0.12s

Presolved: 27 rows, 26 columns, 93 nonzeros

Variable types: 0 continuous, 26 integer (16 binary)

Root relaxation: objective 6.600000e+01, 19 iterations, 0.17 seconds

Nodes			Current Node			Objective Bounds			Work	
Expl	Unexpl		Obj	Depth	IntInf	Incumbent	BestBd	Gap	It/Node	Time
*	0	0			0	66.0000000	66.00000	0.00%	-	0s

Explored 0 nodes (19 simplex iterations) in 0.51 seconds

Thread count was 8 (of 8 available processors)

Solution count 2: 66 72

Optimal solution found (tolerance 1.00e-05)

Best objective 6.600000000000e+01, best bound 6.600000000000e+01, gap 0.0000%

User-callback calls 73, time in user-callback 0.03 sec

Status = nothing

Resultscount = 2

bestsol = 66.0

Elapsed = 7.3455915

Solver time = 0.5140609741210938

y

Nodes		Current Node			Objective Bounds			Work	
Expl	Unexpl	Obj	Depth	IntInf	Incumbent	BestBd	Gap	It/Node	Time
*	0	0		0	66.00000000	66.000000	0.00%	-	0s

Explored 0 nodes (19 simplex iterations) in 0.51 seconds
Thread count was 8 (of 8 available processors)

Solution count 2: 66 72

Optimal solution found (tolerance 1.00e-05)

Best objective 6.600000000000e+01, best bound 6.600000000000e+01, gap 0.0000%

User-callback calls 73, time in user-callback 0.03 sec

Status = nothing

Resultscout = 2

bestsol = 66.0

Elapsed = 7.3455915

Solver time = 0.5140609741210938

y

1,4 = 1

2,2 = 1

3,3 = 1

4,1 = 1

b

0.0

8.0

24.0

38.0

f

8.0

24.0

38.0

58.0

bc

8.0

24.0

38.0

58.0

fc

18.0

32.0

48.0

66.0

BRKGA

Exemplo de codificação

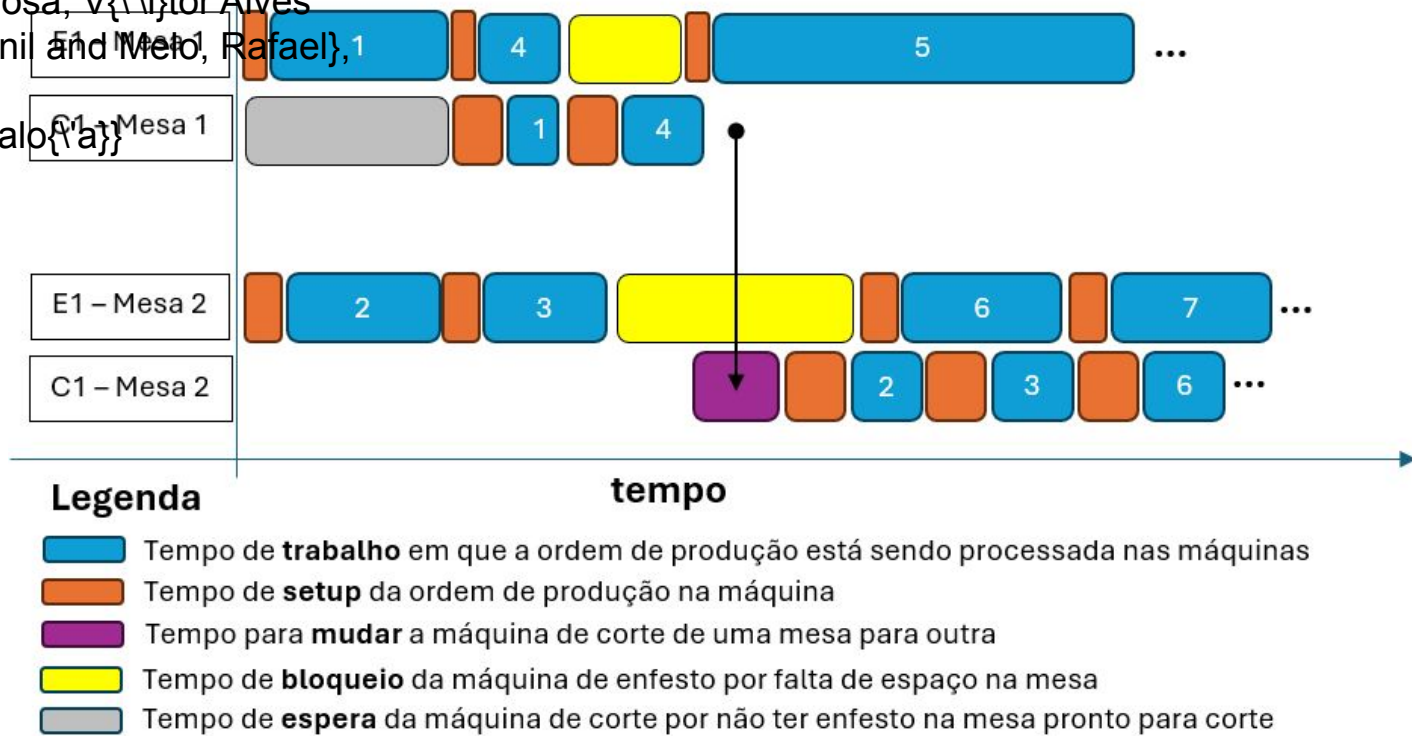
1	2	3	4	5	6	7	8
0.23	0.38	0.11	0.68	0.48	0.97	0.62	0.03

Resultado do BRKGA

```
mateuscsilva@LAPTOP-1EM717S1:/mnt/c/Users/mateu/Documents/Estudo/Github_Repositories/fiesc/case_fiesc/brkga$ ./brkga-main
4
Running for 2147483647 generations using 1 out of 8 available thread units...
  1) Improved best solution thus far: 66
Fitness of the top 10 individuals of each population:
Population #0:
  0) 66
  1) 66
  2) 66
  3) 66
  4) 66
  5) 66
  6) 66
  7) 66
  8) 66
  9) 66
BRKGA run finished in 10.0001 s.
```

Etapa 3

```
@article{barbosa2023problema,  
  title={O problema de coleta e  
entrega com janelas de tempo e  
escalonamento nas arestas},  
  author={Barbosa, V{\i}tor Alves  
and Tiwari, Sunil and Melo, Rafael},  
  year={2023},  
  publisher={Galo{\a}}
```



Exemplo fornecido pela FIESC

Modelo

Modelo

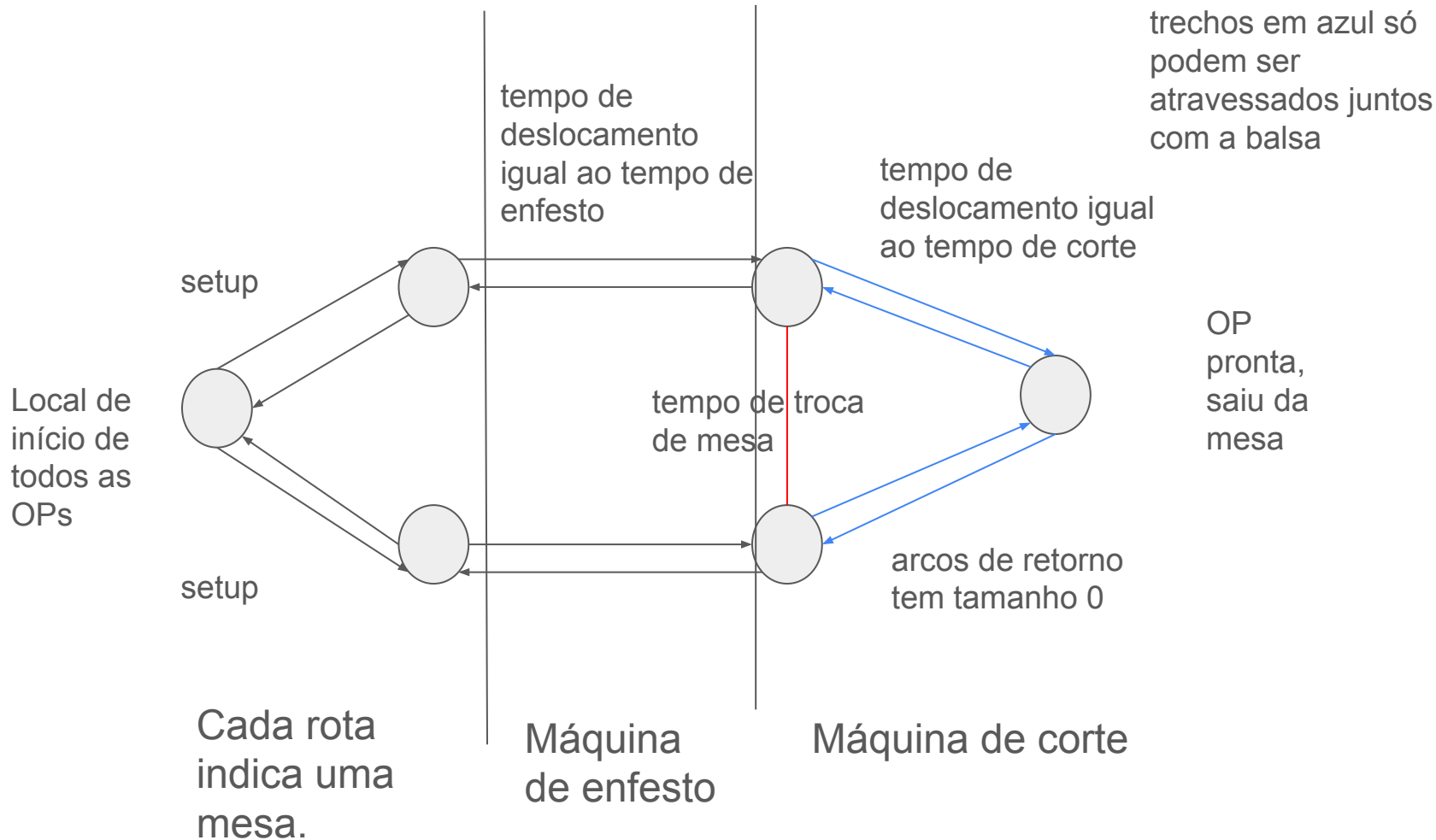
Modelo de programação lógica por restrições

- Garante todas as restrições do problema, exceto a relacionada ao tempo de troca de mesa da máquina de corte.
- Não foi completamente implementada devido a problemas com o pacote ConditionalJuMP que seria utilizado para as duas restrições lógicas
- Código pode ser encontrado no github.

Modelo: abordagem alternativa

Abordar como um problema de roteamento e escalonamento de veículos capacitados.

- Dois veículos capacitados (seriam as mesas)
- Pedidos (OP) que saem de um depósito
- Cada veículo segue uma rota própria (máquina de enfeito)
- O veículo se desloca para um local 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.



Modelo: abordagem alternativa

Uma versão mais simples do [trabalho](#).

- A vantagem é poder ser desenvolvido inteiramente com programação inteira.
- 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 enfeito, corte e capacidade fixa.
- Outra possibilidade poderia ser estender o trabalho: [An approximation algorithm for scheduling two parallel machines with capacity constraints](#), que não conta com *setup*, e algo como uma máquina de corte.

BRKGA

Exemplo de codificação

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

Resultado preliminar

```
mateuscsliva@LAPTOP-1EM717S1:/mnt/c/Users/mateu/Documents/Estudo/Github_Repositories/fiesc/case_fiesc/brkga$ ./brkga-main
9
Running for 2147483647 generations using 1 out of 8 available thread units...
  1) Improved best solution thus far: 1000
  6) Improved best solution thus far: 265
  7) Improved best solution thus far: 253
 11) Improved best solution thus far: 244
 12) Improved best solution thus far: 241
Fitness of the top 10 individuals of each population:
Population #0:
  0) 241
  1) 241
  2) 241
  3) 241
  4) 241
  5) 241
  6) 241
  7) 241
  8) 241
  9) 241
BRKGA run finished in 10.0003 s.
```


Obrigado!