



harve
escola de tecnologia



Pesquisa Operacional aplicada na cadeia de suprimento florestal

APRESENTAÇÃO PESSOAL



Consulting Specialist

AlphaSights · Freelance

Jul 2023 · 1 mo

London, England, United Kingdom · Remote

- Consultancy in forestry software;
- Evaluation criteria and applicability in forest management, control and optimization tools;

Skills: RPA · Technology Integration · Data Management



Forestry Planning and Control Engineer

Bracell · Full-time

Jan 2020 - Jul 2023 · 3 yrs 7 mos

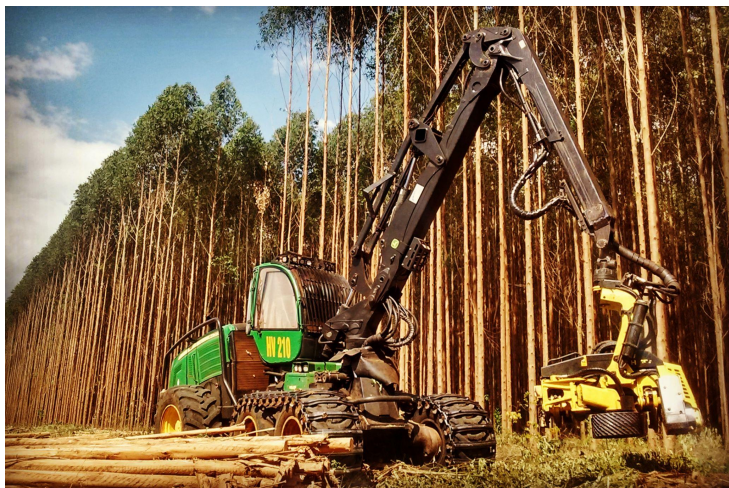
Lençóis Paulista e Região, Brasil · On-site

- Operational and tactical planning of forestry harvesting and logistics;
- Analysis, control and reports of production and financial KPIs;...

Skills: Demand Planning · Technology Integration · Optimization · Data Management



Forestry Planning Team SP
Team Building - 2022



APRESENTAÇÃO PESSOAL



Project Supervisor

Suzano · Full-time

Jan 2019 - Dec 2019 · 1 yr

Tres Lagoas, MS, Brazil · On-site

- Project management and operational development;
- Planning and execution of forest project schedules;...

Skills: Project Management · Operational Development · Technology Integration



Harvest Team

Cutting Material Project - 2019



Fibria

Full-time · 3 yrs 9 mos

Forestry Harvest Supervisor

Apr 2017 - Dec 2018 · 1 yr 9 mos

Três Lagoas, Mato Grosso do Sul, Brazil · On-site

- Crew and safety management;
- Control, monitoring and approval of contracted or outsourced services;...

Skills: Forestry Operations · People Management



Harvest Team MS

Horizon 2 Project - 2017



OBJETIVO DO PROJETO

Criar um código de otimização utilizando a biblioteca *CVXPY para agendamento de operações florestais (baldeio), por meio da maximização de estoque de pilha em períodos chuvosos, restrições de rendimentos e volume disponível.

**“Python-embedded modeling language for convex optimization problems”*

CADEIA DE ABASTECIMENTO FLORESTAL



BANCO DE DADOS

CADASTRO

FAZENDAS	CARACTERÍSTICAS	VOLUME (m³)	MÓDULOS	DATA CORTE	DATA TRANSPORTE
A	Tipo de Acesso Distância / Área Tipo Contratual Localização Floresta (Idade, Material Gen. Espécie, Clone)	10.000	1	JANEIRO	MARÇO
B		20.000	1	JANEIRO	ABRIL
C		35.000	1	FEVEREIRO	MAIO
D		5.000	2	MARÇO	ABRIL
E		30.000	2	MARÇO	MAIO

TABELA DE RENDIMENTOS




MÓDULOS	INDICADORES	Nº DE MÁQUINAS	CAPACIDADE (m³)	DATA BALDEIO
1	Produtividade Efic. Oper. Disp. Mecânica Learning Curve	5	20.000	FEVEREIRO
1		6	30.000	MARÇO
1		4	15.000	ABRIL

RESTRIÇÕES

CADASTRO

FAZENDAS	CARACTERÍSTICAS	VOLUME (m³)	MÓDULOS	DATA CORTE	DATA TRANSPORTE	
A	Acesso Distância / Área Contratos Localização Floresta	10.000	1	JANEIRO	MARÇO	10.000
B		20.000	1	JANEIRO	ABRIL	10.000 + 10.000
C		35.000	1	FEVEREIRO	MAIO	20.000 + 15.000

TABELA DE RENDIMENTOS

MÓDULOS	INDICADORES	Nº DE MÁQUINAS	CAPACIDADE (m³)	DATA BALDEIO
1	Produtividade Efic. Oper. Disp. Mecânica Learning Curve	5	20.000 	FEVEREIRO
1		6	30.000 	MARÇO
1		4	15.000 	ABRIL

harve

FUNÇÃO OBJETIVO

CADASTRO

FAZENDAS	CARACTERÍSTICAS	VOLUME (m³)	MÓDULOS	DATA CORTE	DATA TRANSPORTE	
A	Acesso Distância / Área Contratos Localização Floresta	10.000	1	JANEIRO	MARÇO	10.000
B		20.000	1	JANEIRO	ABRIL	10.000 + 10.000
C		35.000	1	FEVEREIRO	MAIO	20.000 + 15.000

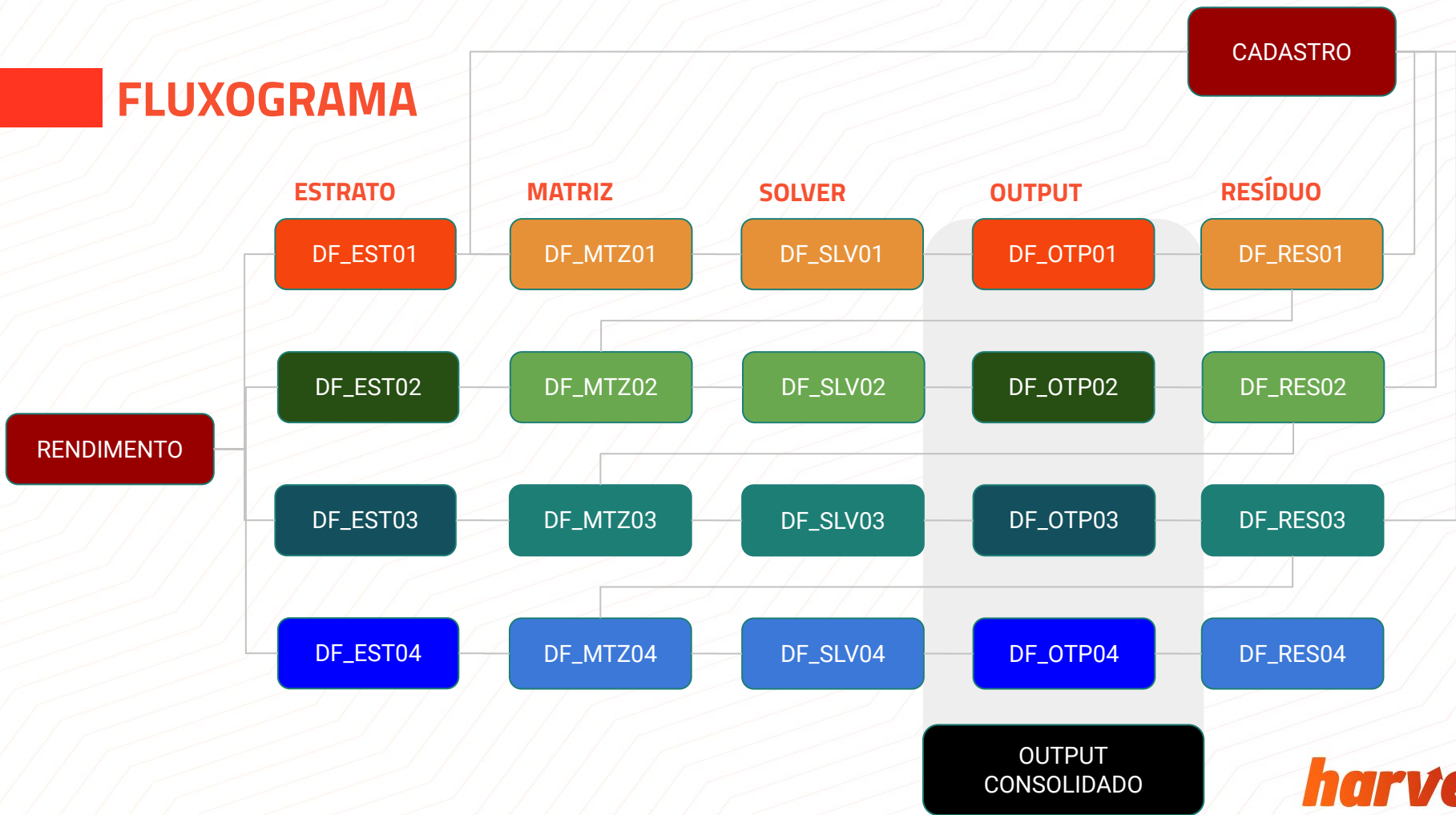
MATRIZ DE VARIÁVEIS - MAXIMIZAÇÃO

FAZENDA	DEZEMBRO	JANEIRO	FEVEREIRO	MARÇO	ABRIL	MAIO	JUNHO
A	-10	+100	+10	+1	-10	-100	-1000
B	-10	+1000	+100	+10	+1	-10	-100
C	-100	-10	+1000	+100	+10	-1	-10

OUTPUT

FAZENDAS	MÓDULOS	JANEIRO	FEVEREIRO	MARÇO	ABRIL	TOTAL
A	1	-	10.000			10.000
B	1	-	10.000	10.000		20.000
C	1	-		20.000	15.000	35.000
TOTAL	1	-	20.000	30.000	15.000	65.000

FLUXOGRAMA



ESTRATIFICAÇÃO DE DATAFRAMES

```
# Obter valores únicos da coluna 'Estrato'
matriz_estratos = df_rend['ID Estrato'].unique()

# Criar novos DataFrames para cada valor único e identificar o nome dos DataFrames
for valor in matriz_estratos:
    # Criar o nome do DataFrame usando o valor único
    nome_do_dataframe = f'df_{valor}'

    # Criar o DataFrame específico para o valor único
    locals()[nome_do_dataframe] = df_rend[df_rend['ID Estrato'] == valor].reset_index(drop=True)

    # Exibir o DataFrame e seu nome
    conj_matriz_estr = f"{nome_do_dataframe}:\n{locals()[nome_do_dataframe]}\n"
    print(conj_matriz_estr)
```


FUNÇÃO OBJETIVO | MATRIZ

```
# Equação de otimização para priorização de dias de pilha e alocação de estoque por classe e sazonalidade
df_0_exp['Sazonalidade'] = df_0_exp['Data Rend'].map(df_sazon.set_index('Data Sazon')['Classe Sazon'])

df_0_exp['Delta Eito'] = df_0_exp['DataMes Rend'] - df_0_exp['Data Corte']
df_0_exp['Delta Pilha'] = df_0_exp['Data Transporte'] - df_0_exp['DataMes Rend']

df_0_exp['Float Eito'] = df_0_exp['Delta Eito'].dt.total_seconds() / (24 * 60 * 60)
df_0_exp['Float Pilha'] = df_0_exp['Delta Pilha'].dt.total_seconds() / (24 * 60 * 60)

# Criar a nova coluna 'inicio_mes'
df_0_exp['Mês Transporte'] = df_0_exp['Data Transporte'].dt.to_period('M').dt.to_timestamp()

# Aplicar uma fórmula com base numa condição
df_0_exp['Valor Otm'] = np.where(df_0_exp['Data Rend'] == df_0_exp['Mês Transporte'],
                                (df_0_exp['Float Eito']*1000000) * df_0_exp['Float Pilha'],
                                np.where(((df_0_exp['Sazonalidade'] == 'Chuva') & (df_0_exp['Classe Estoque'] == 'Chuva')),
                                df_0_exp['Float Eito'] * (df_0_exp['Float Pilha']*1000000),
                                df_0_exp['Float Eito'] * df_0_exp['Float Pilha']))

df_0_exp.to_excel('/content/df_0_exp.xlsx', index=False)
print(df_0_exp)

# Usar pivot para transformar linhas em colunas
df_0_pivot = df_0_exp.pivot(index='Talhão', columns='DataMes Rend', values='Valor Otm').reset_index()
df_0_pivot = df_0_pivot.drop('Talhão', axis=1)
```

RESTRIÇÕES

```
# Definição de restrição de volume=máquinas para a lista de talhões da matriz
df_0_restrtal = df_0_exp[['Talhão', 'VolTal Máq']]
df_0_restrtal = df_0_restrtal.drop_duplicates(subset='Talhão')
print(df_0_restrtal)

# Ordenar os talhões para coincidir com a sequência pivot
df_0_restrtal = df_0_restrtal.sort_values(by=['Talhão'], ascending=[True])
print(df_0_restrtal)

# Definição de restrição de máquinas para o intervalo de dias da matriz
df_0_restrmaq = df_0_exp[['DataMes Rend', 'Máquinas']]
df_0_restrmaq = df_0_restrmaq.drop_duplicates(subset='DataMes Rend')
print(df_0_restrmaq)
```

SOLVER

```
df_0_bpc_dias = df_0_pivot.to_numpy()
df_0_oferta_tal = df_0_restrtal['VolTal Máq'].to_numpy()
df_0_demanda_maq = df_0_restrmaq['Máquinas'].to_numpy()

print(df_0_bpc_dias)
print(df_0_oferta_tal)
print(df_0_demanda_maq)

# DEFINIÇÃO DOS EIXOS DA MATRIZ DE VARIÁVEIS
df_0_ei = np.ones(df_0_bpc_dias.shape[0])
df_0_ej = np.ones(df_0_bpc_dias.shape[1])

print(df_0_ei.shape)
print(df_0_ej.shape)

# DEFINIÇÃO DAS RESTRIÇÕES
lower = np.zeros(df_0_bpc_dias.shape)
df_0_x = cp.Variable(df_0_bpc_dias.shape)
df_0_bpc0 = df_0_x@df_0_ej <= df_0_oferta_tal
df_0_bpc1 = cp.transpose(df_0_x)@df_0_ei <= df_0_demanda_maq
df_0_bpc2 = df_0_x >= lower

# FUNÇÃO OBJETIVO
df_0_obj = cp.multiply(df_0_bpc_dias,df_0_x)@df_0_ej@df_0_ei
```

```
# SOLVER
df_0_prob = cp.Problem(cp.Maximize(df_0_obj), [df_0_bpc0, df_0_bpc1, df_0_bpc2])
df_0_solv = df_0_prob.solve(solver=cp.GLPK_MI)

print(df_0_solv)
print("status:",df_0_prob.status)
print('restrição talhão:', np.matmul(df_0_x.value,df_0_ej))
print('restrição máquinas:', np.matmul(np.transpose(df_0_x.value),df_0_ei))
print("Valor ótimo de x:", df_0_x.value)
```


DASHBOARD

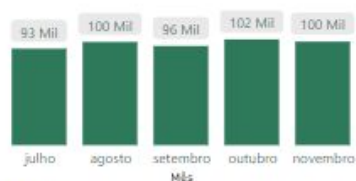
Módulo

- MD01
- MD02
- MD03

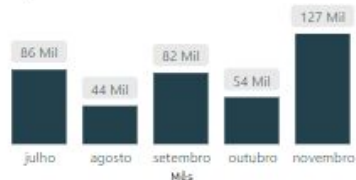
Corte



Baldeio



Transporte

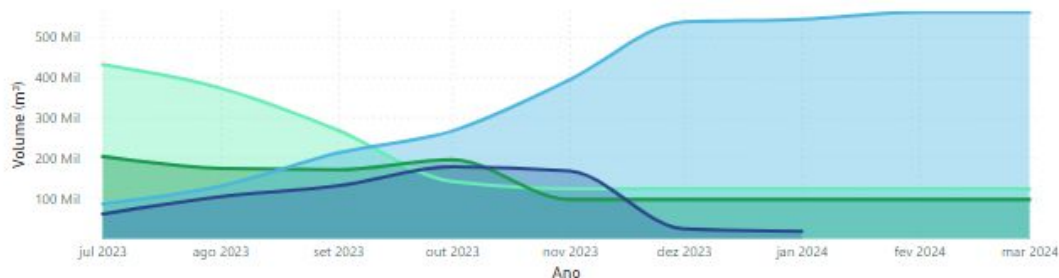


Chuva Seco



Visão de Estoques

Posição da Madeira Floresta Em Pé Madeira Eito Madeira Pilha Transportado



Liberção de área (ha)

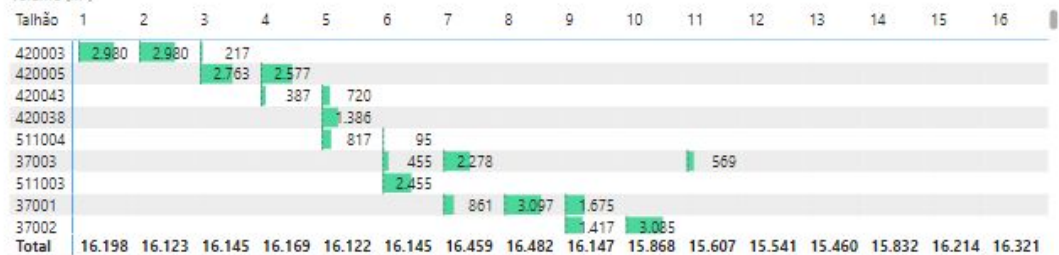


Próprio Parceria Rural Compra de Madeira



Sequenciamento Operacional

Volume (m³)



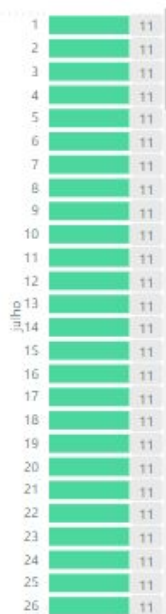
harve

DASHBOARD

● FULL



N° de Máquinas



Módulo

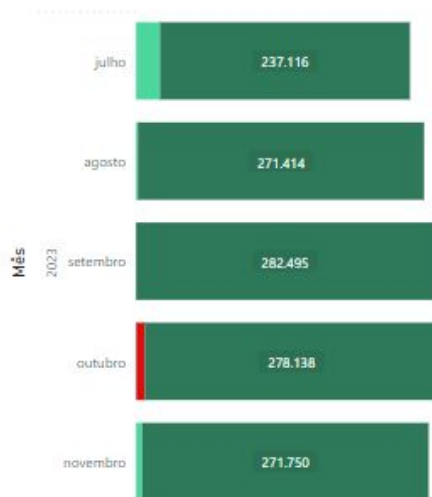
☐ MD01

☐ MD02

☐ MD03

Baldeio Mensal

Status Oper ● Andamento ● Inconsistente ● Pilha



Soma de Sched Vol (m³)

EITO

Ano	julho	agosto	setembro	outubro	novembro	Total
2023	259.796	273.082	282.495	286.520	277.488	1.379.381
janeiro	6.177	23.141				29.318
fevereiro	5.340	10.184				15.524
março	1.774	30				1.804
abril	2.493	381				2.874
maio	61.747	20.922				82.669
junho	160.278	49.395	96.278			305.951
julho	21.987	123.457	46.311			191.755
agosto		45.572	91.862	69.190		206.624
setembro			48.044	61.410	147.245	256.699
outubro				147.655	67.011	214.666
novembro				2.996	63.232	66.228
Total	259.796	273.082	282.495	286.520	277.488	1.379.381

PILHA

Ano	julho	agosto	setembro	outubro	novembro	Total
2023	259.796	273.082	282.495	265.333	208.039	1.288.745
julho	166.513					166.513
agosto	8.133	104.053				112.186
setembro	63.403	79.865	211.598			354.866
outubro	17.410	56.180	7.714	126.886		208.190
novembro	4.337	32.984	40.344		184.905	262.570
dezembro			22.839	138.447	23.134	184.420
2024				21.187	69.449	90.636
janeiro				21.187	44.333	65.520
fevereiro					25.116	25.116
Total	259.796	273.082	282.495	286.520	277.488	1.379.381

harve

CONCLUSÃO FINAL E PRÓXIMOS PASSOS

- ETL a partir de uma tabela de input padrão
- Modelagem das variáveis de otimização e aplicação CVXPY
- Estruturação e fluxograma do código
- Regras para validação de consistência do modelo
- Dashboard de visualização de cenários otimizados Power BI
- **Automatização do código para atualizações do banco de dados**