

redes_neurais

September 30, 2019



Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina

Título: Algoritmo Evolucionário Multi-Objetivo aplicado ao Problema de Despacho de Sistemas Elétricos

Professor: Dr. Eng. Sergio Luciano Avila

Acadêmicos: Fábio Matheus Mantelli e Douglas Pereira Dias

0.0.1 Justificativa do trabalho:

Usinas termelétricas são os empreendimentos que mais poluem a atmosfera por meio da emissão de óxidos de enxofre (SOx) e óxidos de nitrogênio (NOx). Atualmente, o problema de minimização de poluição tem ganhado grande destaque e requer medidas para manter o ar das grandes cidades mais limpo. Além disso, o aumento da conscientização pública sobre a proteção ambiental e a aprovação de leis força as concessionárias a modificar os seus projetos e as estratégias operacionais com o objetivo de reduzir a poluição e as emissões de gases à atmosfera.

Algumas estratégias foram propostas na literatura, tais como:

- Instalação de equipamentos (precipitadores eletrostáticos);

- Mudança para combustíveis de baixa emissão;

- Substituição de unidades geradoras antigas e ineficientes;

- Despacho de emissão. As três primeiras alternativas requerem a instalação de novos equipamentos ou modificação dos existentes. Porém, há um grande gasto de capital a ser investido para a substituição, portanto, podem apenas ser consideradas a longo prazo. Por outro lado, a opção de despacho de emissões é uma alternativa de curto prazo, na qual a emissão, além do custo do combustível, podem ser minimizados.

Com uma nova função objetivo (reduzir a emissão de gases poluentes), o problema de despacho econômico pode ser tratado como um problema de otimização multiobjetivo com objetivos contraditórios, uma vez que a solução ótima do problema de despacho econômico não é a melhor solução ambiental.

0.0.2 Sistema IEEE 30 barras e 6 geradores termelétricos

No presente estudo, o problema de EED (Despacho Econômico/Ambiental, em tradução livre) é formulado como um problema de otimização multi-objetivo com restrição não-linear, onde o custo do combustível e o impacto ambiental são tratados como objetivos conflitantes. O algoritmo NSGA (Non-dominated Sort Genetic Algorithm) é aplicado a um sistema teste IEEE 30 barras e com 6 geradores termelétricos. As perdas nas linhas de transmissão são desconsideradas. Na Figura abaixo é possível visualizar o sistema IEEE 30 barras.

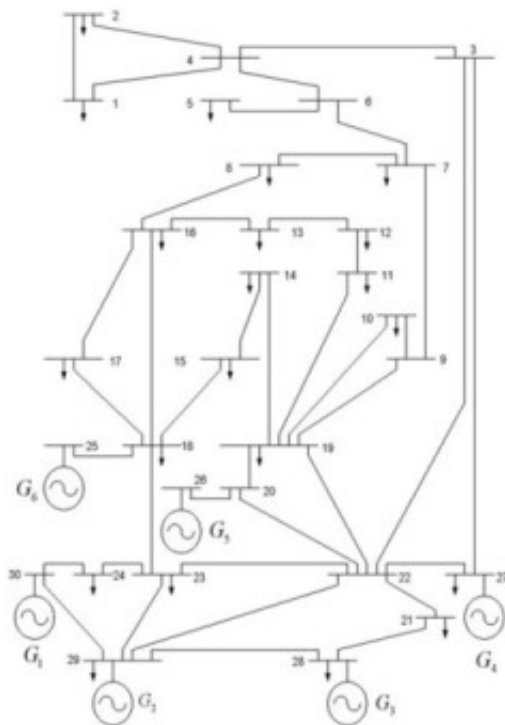


Figura 1 - Sistema teste IEEE 30 barras com 6 geradores

0.0.3 Dados dos Geradores

Sistema IEEE com 30 barras e 6 geradores parâmetros: <https://rwl.github.io/PYPOWER/api/pypower.case30-pysrc.html>

```
[11]: from pypower.api import case30, ppoption, runpf, printpf
import numpy as np

# utiliza o sistema IEEE de 30 barras e 6 geradores
ppc = case30()
ppopt = ppoption(PF_ALG=1)
r = runpf(ppc, ppopt)
```

```

# calcula o somatório de potência dos geradores
potencia = 0
for i in range(0, len(r[0]['gen'][:])):
    potencia += r[0]['gen'][i][1]

# calcula o somatório da demanda das cargas
demanda = 0
for i in range(0, len(r[0]['bus'])):
    demanda += r[0]['bus'][i][2]

```

PYPOWER Version 5.1.4, 27-June-2018 -- AC Power Flow (Newton)

Newton's method power flow converged in 3 iterations.

Converged in 0.04 seconds

```

=====
|      System Summary                                     |
=====

```

How many?		How much?	P (MW)	Q (MVar)
Buses	30	Total Gen Capacity	335.0	-95.0 to 405.9
Generators	6	On-line Capacity	335.0	-95.0 to 405.9
Committed Gens	6	Generation (actual)	191.6	100.4
Loads	20	Load	189.2	107.2
Fixed	20	Fixed	189.2	107.2
Dispatchable	0	Dispatchable	0.0 of 0.0	0.0
Shunts	2	Shunt (inj)	-0.0	0.2
Branches	41	Losses ($I^2 * Z$)	2.44	8.99
Transformers	0	Branch Charging (inj)	-	15.6
Inter-ties	7	Total Inter-tie Flow	33.2	27.1
Areas	3			

	Minimum	Maximum
Voltage Magnitude	0.961 p.u. @ bus 8	1.000 p.u. @ bus 1
Voltage Angle	-3.96 deg @ bus 19	1.48 deg @ bus 13
P Losses ($I^2 * R$)	-	0.29 MW @ line 2-6
Q Losses ($I^2 * X$)	-	2.10 MVar @ line 12-13

```

=====
|      Bus Data                                           |
=====

```

Bus	Voltage		Generation		Load	
#	Mag(pu)	Ang(deg)	P (MW)	Q (MVar)	P (MW)	Q (MVar)

1	1.000	0.000*	25.97	-1.00	-	-
2	1.000	-0.415	60.97	32.00	21.70	12.70
3	0.983	-1.522	-	-	2.40	1.20
4	0.980	-1.795	-	-	7.60	1.60
5	0.982	-1.864	-	-	-	-
6	0.973	-2.267	-	-	-	-
7	0.967	-2.652	-	-	22.80	10.90
8	0.961	-2.726	-	-	30.00	30.00
9	0.981	-2.997	-	-	-	-
10	0.984	-3.375	-	-	5.80	2.00
11	0.981	-2.997	-	-	-	-
12	0.985	-1.537	-	-	11.20	7.50
13	1.000	1.476	37.00	11.35	-	-
14	0.977	-2.308	-	-	6.20	1.60
15	0.980	-2.312	-	-	8.20	2.50
16	0.977	-2.644	-	-	3.50	1.80
17	0.977	-3.392	-	-	9.00	5.80
18	0.968	-3.478	-	-	3.20	0.90
19	0.965	-3.958	-	-	9.50	3.40
20	0.969	-3.871	-	-	2.20	0.70
21	0.993	-3.488	-	-	17.50	11.20
22	1.000	-3.393	21.59	39.57	-	-
23	1.000	-1.589	19.20	7.95	3.20	1.60
24	0.989	-2.631	-	-	8.70	6.70
25	0.990	-1.690	-	-	-	-
26	0.972	-2.139	-	-	3.50	2.30
27	1.000	-0.828	26.91	10.54	-	-
28	0.975	-2.266	-	-	-	-
29	0.980	-2.128	-	-	2.40	0.90
30	0.968	-3.042	-	-	10.60	1.90
Total:			191.64	100.41	189.20	107.20

Branch Data								
Brnch #	From Bus	To Bus	From Bus P (MW)	Injection Q (MVar)	To Bus P (MW)	Injection Q (MVar)	Loss ($I^2 * Z$)	
							P (MW)	Q (MVar)
0	1	2	10.89	-5.09	-10.86	2.17	0.026	0.08
1	1	3	15.08	4.09	-14.96	-5.57	0.127	0.48
2	2	4	16.07	5.21	-15.89	-6.66	0.178	0.50
3	3	4	12.56	4.37	-12.54	-4.30	0.018	0.07
4	2	5	13.79	4.51	-13.68	-6.03	0.110	0.44
5	2	6	20.28	7.42	-19.99	-8.50	0.289	0.87
6	4	6	22.50	11.38	-22.43	-11.12	0.066	0.26
7	5	7	13.68	6.21	-13.56	-6.88	0.120	0.29

8	6	7	9.27	3.17	-9.24	-4.02	0.031	0.08
9	6	8	24.82	24.43	-24.69	-23.92	0.128	0.51
10	6	9	5.79	-3.36	-5.79	3.46	-0.000	0.10
11	6	10	3.31	-1.92	-3.31	2.00	-0.000	0.09
12	9	11	0.00	0.00	0.00	0.00	-0.000	0.00
13	9	10	5.79	-3.46	-5.79	3.51	-0.000	0.05
14	4	12	-1.67	-2.02	1.67	2.04	-0.000	0.02
15	12	13	-37.00	-9.26	37.00	11.35	-0.000	2.10
16	12	14	5.39	0.88	-5.35	-0.80	0.037	0.08
17	12	15	9.48	-1.06	-9.41	1.19	0.066	0.12
18	12	16	9.26	-0.10	-9.18	0.28	0.080	0.18
19	14	15	-0.85	-0.80	0.85	0.80	0.003	0.00
20	16	17	5.68	-2.08	-5.65	2.15	0.031	0.07
21	15	18	9.16	0.76	-9.07	-0.57	0.097	0.19
22	18	19	5.87	-0.33	-5.85	0.38	0.022	0.05
23	19	20	-3.65	-3.78	3.66	3.80	0.009	0.02
24	10	20	5.92	4.62	-5.86	-4.50	0.052	0.12
25	10	17	3.37	8.01	-3.35	-7.95	0.023	0.06
26	10	21	-2.23	-11.67	2.28	11.77	0.044	0.10
27	10	22	-3.75	-8.48	3.82	8.62	0.062	0.13
28	21	22	-19.78	-22.97	19.87	23.16	0.093	0.19
29	15	23	-8.81	-5.25	8.91	5.47	0.109	0.22
30	22	24	-2.10	7.80	2.18	-7.68	0.078	0.12
31	23	24	7.09	0.88	-7.02	-0.75	0.066	0.14
32	24	25	-3.86	1.77	3.89	-1.71	0.035	0.06
33	25	26	3.55	2.37	-3.50	-2.30	0.046	0.07
34	25	27	-7.44	-0.66	7.50	0.78	0.063	0.12
35	28	27	-6.11	-6.08	6.11	6.40	-0.000	0.31
36	27	29	6.17	1.68	-6.08	-1.51	0.090	0.17
37	27	30	7.12	1.67	-6.95	-1.35	0.171	0.32
38	29	30	3.68	0.61	-3.65	-0.55	0.035	0.07
39	8	28	-5.31	-6.08	5.34	4.33	0.036	0.12
40	6	28	-0.77	-2.70	0.77	1.75	0.001	0.00
Total:							2.444	8.99

Coeficientes de Emissão de COx e de Custos dos Geradores

```
[12]: import math
from mpmath import mp
mp.dps = 300

# parâmetros do gerador
gerador = [
    {
        "Cost": [10, 200, 100],
        "Emission": [4.091, -5.554, 6.490, 20E-4, 2.857]
    },

```

```

{
  "Cost": [10, 150, 120],
  "Emission": [2.543, -6.047, 5.638, 5.0E-4, 3.333]
},
{
  "Cost": [20, 180, 40],
  "Emission": [4.258, -5.094, 4.586, 1.0E-6, 8.000]
},
{
  "Cost": [10, 100, 60],
  "Emission": [5.326, -3.550, 3.380, 2.0E-3, 1.0E-6, 1.0E-5]
},
{
  "Cost": [20, 180, 40],
  "Emission": [4.258, -5.094, 4.586, 1.0E-6, 8.000]
},
{
  "Cost": [10, 150, 100],
  "Emission": [6.131, -5.555, 5.151, 1.0E-5, 6.667]
}
]

```

0.0.4 IEEE 30-bus

Demanda de potência ativa e reativa pelas cargas PD (MW) | QD (MVar) |

```

[13]: bus_data = [
    [0.00, 0.00],
    [21.70, 12.70],
    [2.40, 120],
    [7.60, 1.60],
    [94.20, 19.00],
    [0.00, 0.00],
    [22.80, 10.90],
    [30.00, 30.00],
    [0.00, 0.00],
    [5.80, 2.00],
    [0.00, 0.00],
    [11.20, 7.50],
    [0.00, 0.00],
    [6.20, 1.60],
    [8.20, 2.50],
    [3.50, 1.80],
    [9.00, 5.80],
    [3.20, 0.90],
    [9.50, 3.40],
    [2.20, 0.70],

```

```

[17.50, 11.20],
[0.00, 0.00],
[3.20, 1.60],
[8.70, 6.70],
[0.00, 0.00],
[3.50, 2.30],
[0.00, 0.00],
[0.00, 0.00],
[2.40, 0.90],
[10.60, 1.90]
]

```

0.0.5 IEEE 30-bus

Dados das barras e linhas Line | From Bus | To Bus | Resistance | Reactance | Susceptance | Rating (MVA) |

```

[14]: line_data = [
    [ 1,  2, 0.0192, 0.0575, 0.0264, 130],
    [ 1,  3, 0.0452, 0.1852, 0.0204, 130],
    [ 2,  4, 0.0570, 0.1737, 0.0184,  65],
    [ 3,  4, 0.0132, 0.0379, 0.0042, 130],
    [ 2,  5, 0.0472, 0.1983, 0.0209, 130],
    [ 2,  6, 0.0581, 0.1763, 0.0187,  65],
    [ 4,  6, 0.0119, 0.0414, 0.0045,  90],
    [ 5,  7, 0.0460, 0.1160, 0.0102,  70],
    [ 6,  7, 0.0267, 0.0820, 0.0085, 130],
    [ 6,  8, 0.0120, 0.0420, 0.0045,  32],
    [ 6,  9, 0.0000, 0.2080, 0.0000,  65],
    [ 6, 10, 0.0000, 0.5560, 0.0000,  32],
    [ 9, 11, 0.0000, 0.2080, 0.0000,  65],
    [ 9, 10, 0.0000, 0.1100, 0.0000,  65],
    [ 4, 12, 0.0000, 0.2560, 0.0000,  65],
    [12, 13, 0.0000, 0.1400, 0.0000,  65],
    [12, 14, 0.1231, 0.2559, 0.0000,  32],
    [12, 15, 0.0662, 0.1304, 0.0000,  32],
    [12, 16, 0.0945, 0.1987, 0.0000,  32],
    [14, 15, 0.2210, 0.1997, 0.0000,  16],
    [16, 17, 0.0824, 0.1923, 0.0000,  16],
    [15, 18, 0.1070, 0.2185, 0.0000,  16],
    [18, 19, 0.0639, 0.1292, 0.0000,  16],
    [19, 20, 0.0340, 0.0680, 0.0000,  32],
    [10, 20, 0.0936, 0.2090, 0.0000,  32],
    [10, 17, 0.0324, 0.0845, 0.0000,  32],
    [10, 21, 0.0348, 0.0749, 0.0000,  32],
    [10, 22, 0.0727, 0.1499, 0.0000,  32],
    [21, 22, 0.0116, 0.0236, 0.0000,  32],

```

```
[15, 23, 0.1000, 0.2020, 0.0000, 16],
[22, 24, 0.1150, 0.1790, 0.0000, 16],
[23, 24, 0.1320, 0.2700, 0.0000, 16],
[24, 25, 0.1885, 0.3292, 0.0000, 16],
[25, 26, 0.2544, 0.3800, 0.0000, 16],
[25, 27, 0.1093, 0.2087, 0.0000, 16],
[28, 27, 0.0000, 0.3960, 0.0000, 65],
[27, 29, 0.2198, 0.4153, 0.0000, 16],
[27, 30, 0.3202, 0.6027, 0.0000, 16],
[29, 30, 0.2399, 0.4533, 0.0000, 16],
[ 8, 28, 0.0636, 0.2000, 0.0214, 32],
[ 6, 28, 0.0169, 0.0599, 0.0065, 32]
```

]

1 Problema de Despacho Econômico/Ambiental

1.0.1

1.1 Problema proposto no livro “Intelligent Systems”

Autores: Christine L. Mumford e Lakhmi C. Jain

1.1.1

1.2 Objetivos do Problema

Minimizar o Custo do Combustível: $F(P_G) = \sum_{i=1}^N a_i + b_i P_{G_i} + c_i P_{G_i}^2$ O vetor da energia ativa dos geradores é definida da seguinte forma: $P_G = [P_{G_1}, P_{G_2}, \dots, P_{G_N}]$

Minimizar a Emissão de SO_x : $E(P_G) = \sum_{i=1}^N 10^{-2} (\alpha_i + \beta_i P_{G_i} + \gamma_i P_{G_i}^2) + \zeta_i e^{\lambda_i P_{G_i}}$ Onde: $\alpha_i, \beta_i, \gamma_i, \zeta_i$ e λ_i são coeficientes das características de emissão do gerador i^{th}

1.3 Restrições do Problema

Restrição de Capacidade de Geração: $P_{G_i}^{min} \leq P_{G_i} \leq P_{G_i}^{max}, i = 1, \dots, N$

Restrições de Balanço de Energia: $\sum_{i=1}^N P_{G_i} - P_D = 0$

```
[15]: # importa bibliotecas do Platypus (Framework de computação evolucionária
      ↪ multiobjetivo em Python)
from platypus import NSGAI, Problem, Real

# Definição da classe otimizacao
class otimizacao(Problem):

    def __init__(self):

        # init (6 variaveis, 2 objetivos, 13 restrições)
```



```

super(otimizacao, self).__init__(6, 2, 13)

# definição dos limites de cada uma das variáveis (são 6 variáveis)
self.types[:] = [Real(0, 10000), Real(0, 10000), Real(0, 10000),
→Real(0, 10000), Real(0, 10000), Real(0, 10000)]

# primeira restrição de igualdade
self.constraints[0:6] = ">=0"

# segunda, terça e quarta restrições definidas como inequações (maior
→ou igual a zero)
self.constraints[6:12] = "<=0"

# demais restrições como inequações (menor ou igual a zero)
self.constraints[12] = "==0"

# função para calcular o problema de otimização multiobjetivo
def evaluate(self, solution):

    # P[1:6] são 6 variáveis do problema
    P = solution.variables[:]

    # calcula o somatório dos 2 objetivos do problema
    min_custo_combustivel = 0
    min_emissao = 0
    for i in range(0, len(gerador)):
        min_custo_combustivel += gerador[i]['Cost'][0] +
→gerador[i]['Cost'][1] * P[i] + gerador[i]['Cost'][2] * P[i] ** 2
        min_emissao += 10 ** (-2) * (gerador[i]['Emission'][0] +
→gerador[i]['Emission'][1] * P[i] + gerador[i]['Emission'][2] * P[i] ** 2) +
→gerador[i]['Emission'][3] * mp.exp(gerador[i]['Emission'][4] * P[i])

    # definição da funções multiobjetivo
    solution.objectives[:] = [ min_custo_combustivel, min_emissao]

    # definição das restrições
    solution.constraints[:] = [P[0], P[1], P[2] -50, P[3] -55 , P[4], P[5],
→P[0] - 80, P[1] - 80, P[2] -1, P[3] -1, P[4] -30, P[5] -40, potencia -
→demanda]

# utiliza o algoritmo NSGAII para otimizar o problema
algorithm = NSGAII(otimizacao())

# otimiza o problema usando 1000 funções de avaliação
funcoes_de_avaliacao = 15000
algorithm.run(funcoes_de_avaliacao)

```

```
# conclui o processo
print("Processo finalizado.")
```

Processo finalizado.

1.4 Define uma tela para imprimir os resultados

```
[16]: print("Resultado da computacao evolucionaria multiobjetivo em Python")
print("\n")
print(f"Numero de funcoes de avaliacao: {funcoes_de_avaliacao}")
print("\n")
print("=====")
print("|\\t\\t\\t\\tSumario \\t\\t\\t\\t|")
print("-----")
print("\\t Funcao Objetivo 1 \\t\\t\\t Funcao Objetivo 2")
print("-----")

for solution in algorithm.result:
    # imprime o resultado da função multiobjetivo
    print('\\t %.4f'%(solution.objectives[0]), '\\t\\t\\t\\t %.4f'%(solution.
    ↪objectives[1]))

print("-----")
print("|\\t\\t\\t\\tCusto dos Geradores \\t\\t\\t\\t|")
print("-----")
print("| \\tP1", "    |", " P2", "    |", " P3", "    |", " P4", "    |", " P5", "    |", " P6\\t\\t|")
print("-----")
for solution in algorithm.result:
    # imprime o resultado das variáveis
    print('\\t%.4f'%(solution.variables[0]), '    %.4f'%(solution.variables[1]), '    ↪
    ↪%.4f'%(solution.variables[2]), '    %.4f'%(solution.variables[3]), '    ↪
    ↪%.4f'%(solution.variables[4]), '    %.4f'%(solution.variables[5]))
```

Resultado da computacao evolucionaria multiobjetivo em Python

Numero de funcoes de avaliacao: 15000

```
=====
=
|                               Sumario
|
-----
-
```

Funcao Objetivo 1	Funcao Objetivo 2
-	
710.6494	0.2251
467.9258	0.2687
679.1528	0.2272
708.1604	0.2267
639.1280	0.2314
667.0301	0.2280
653.2344	0.2297
660.6810	0.2287
608.9800	0.2363
612.6739	0.2354
587.7573	0.2398
571.9063	0.2421
605.6204	0.2375
658.5196	0.2289
709.4922	0.2267
515.8176	0.2548
475.6680	0.2662
488.7172	0.2618
492.2869	0.2610
507.5336	0.2566
619.4559	0.2345
564.1133	0.2440
600.9839	0.2378
580.0501	0.2409
473.6902	0.2669
505.5251	0.2572
606.8645	0.2367
471.3026	0.2679
653.8894	0.2296
479.1933	0.2653
538.5735	0.2490
517.6812	0.2541
494.8081	0.2601
567.8311	0.2430
534.6559	0.2498
483.4049	0.2635
567.1591	0.2436
497.1273	0.2596
558.8420	0.2448
532.2491	0.2506
628.6209	0.2332
589.6860	0.2392
627.2044	0.2339
637.7446	0.2322
498.8701	0.2588

547.6430	0.2469
635.2493	0.2329
486.3548	0.2626
573.0842	0.2418
477.1296	0.2655
630.5683	0.2329
535.7442	0.2495
468.4444	0.2685
523.2992	0.2526
545.2863	0.2474
597.4614	0.2379
490.3567	0.2613
585.2268	0.2406
556.7141	0.2453
597.3937	0.2386
519.5695	0.2537
593.7824	0.2387
623.0656	0.2340
616.6616	0.2348
471.9782	0.2674
480.4653	0.2643
501.0981	0.2583
577.5925	0.2416
480.8773	0.2643
510.2168	0.2561
530.8797	0.2507
591.6349	0.2389
615.2087	0.2351
554.2132	0.2461
512.5242	0.2553
484.3821	0.2632
577.8623	0.2413
526.5780	0.2517
636.2017	0.2324
585.2255	0.2408
528.9665	0.2513
520.9129	0.2531
549.1597	0.2466
503.1850	0.2578
504.0523	0.2578
510.5511	0.2558
485.6109	0.2627
561.2068	0.2445
495.6965	0.2599
525.8784	0.2521
500.5118	0.2587
544.1213	0.2477
524.9898	0.2524

540.2171	0.2486
551.3391	0.2464
624.6278	0.2340
528.6059	0.2514
520.6453	0.2533
512.2624	0.2554
555.1640	0.2455

-

|

Custo dos Geradores

|

P1	P2	P3	P4	P5	P6
0.3123	0.2661	1.0278	1.0032	0.3100	0.3394
0.0000	0.0000	1.0278	1.0023	0.0007	0.0002
0.1656	0.2619	1.0278	1.0146	0.3159	0.3486
0.3139	0.2276	1.0278	1.0228	0.3146	0.3388
0.1141	0.2664	1.0278	1.0032	0.3100	0.2256
0.1154	0.2664	1.0278	1.0022	0.3213	0.3486
0.0901	0.2664	1.0278	1.0032	0.3100	0.3204
0.1141	0.2664	1.0278	1.0032	0.3100	0.3305
0.1566	0.2577	1.0278	1.0023	0.1303	0.2140
0.1512	0.2491	1.0278	1.0027	0.1726	0.2064
0.0172	0.2665	1.0278	1.0027	0.1735	0.2079
0.0551	0.1885	1.0278	1.0036	0.1318	0.2082
0.0284	0.2664	1.0278	1.0022	0.1367	0.3203
0.1141	0.2664	1.0278	1.0032	0.3100	0.3204
0.3170	0.2276	1.0278	1.0228	0.3171	0.3388
0.0181	0.0498	1.0278	1.0021	0.0752	0.1397
0.0025	0.0007	1.0278	1.0027	0.0034	0.0427
0.0038	0.0561	1.0278	1.0029	0.0007	0.0709
0.0210	0.0449	1.0278	1.0027	0.0020	0.0813
0.0046	0.1340	1.0278	1.0027	0.0162	0.0856
0.1297	0.2615	1.0278	1.0022	0.1411	0.2831
0.0214	0.1422	1.0278	1.0028	0.1339	0.2479
0.1566	0.2576	1.0278	1.0023	0.1272	0.1748
0.0423	0.2479	1.0278	1.0023	0.1284	0.2069
0.0084	0.0004	1.0278	1.0027	0.0038	0.0224
0.0119	0.1340	1.0278	1.0027	0.0001	0.0822
0.1566	0.2576	1.0278	1.0023	0.1284	0.2049
0.0131	0.0002	1.0278	1.0027	0.0019	0.0030
0.0934	0.2661	1.0278	1.0032	0.3100	0.3204
0.0190	0.0003	1.0278	1.0027	0.0055	0.0420
0.0154	0.1394	1.0278	1.0022	0.1343	0.1209

0.0158	0.1499	1.0278	1.0027	0.0013	0.1301
0.0064	0.1128	1.0278	1.0027	0.0023	0.0441
0.0561	0.1273	1.0278	1.0027	0.1722	0.2065
0.0217	0.1394	1.0278	1.0022	0.0779	0.1515
0.0114	0.0418	1.0278	1.0027	0.0025	0.0412
0.0214	0.2506	1.0278	1.0036	0.1318	0.1527
0.0179	0.1128	1.0278	1.0027	0.0023	0.0441
0.0542	0.1263	1.0278	1.0023	0.1284	0.2069
0.0216	0.1375	1.0278	1.0027	0.0385	0.1794
0.1154	0.2653	1.0278	1.0032	0.1662	0.3141
0.0365	0.2583	1.0278	1.0027	0.1719	0.2079
0.1571	0.1867	1.0278	1.0146	0.2543	0.2426
0.1154	0.2664	1.0278	1.0032	0.1736	0.3486
0.0041	0.1060	1.0278	1.0029	0.0006	0.0810
0.0332	0.1499	1.0278	1.0022	0.1254	0.1501
0.1558	0.2577	1.0278	1.0022	0.1297	0.3434
0.0114	0.0572	1.0278	1.0027	0.0025	0.0441
0.0591	0.1891	1.0278	1.0022	0.1296	0.2132
0.0015	0.0522	1.0278	1.0027	0.0004	0.0050
0.1154	0.2686	1.0278	1.0027	0.1662	0.3204
0.0182	0.1287	1.0278	1.0022	0.0890	0.1608
0.0000	0.0025	1.0278	1.0023	0.0007	0.0011
0.0202	0.1368	1.0278	1.0022	0.0836	0.0845
0.0217	0.1394	1.0278	1.0022	0.1343	0.1515
0.0544	0.2765	1.0278	1.0027	0.1727	0.2079
0.0038	0.0561	1.0278	1.0029	0.0006	0.0810
0.0346	0.1654	1.0278	1.0021	0.1367	0.3143
0.0552	0.1273	1.0278	1.0027	0.1726	0.1471
0.1537	0.2576	1.0278	1.0036	0.1318	0.1527
0.0181	0.1287	1.0278	1.0027	0.0003	0.1608
0.0369	0.2765	1.0278	1.0027	0.1727	0.2079
0.1297	0.2615	1.0278	1.0022	0.1411	0.3005
0.1621	0.2503	1.0278	1.0027	0.1727	0.2127
0.0000	0.0002	1.0278	1.0027	0.0038	0.0224
0.0005	0.0560	1.0278	1.0022	0.0057	0.0187
0.0151	0.1060	1.0278	1.0029	0.0006	0.0810
0.0156	0.2506	1.0278	1.0036	0.1318	0.2150
0.0019	0.0560	1.0278	1.0027	0.0057	0.0187
0.0290	0.1375	1.0278	1.0027	0.0036	0.0818
0.0126	0.1224	1.0278	1.0022	0.0833	0.1524
0.0369	0.2665	1.0278	1.0027	0.1727	0.2079
0.1621	0.2491	1.0278	1.0027	0.1727	0.2064
0.0553	0.1468	1.0278	1.0027	0.1825	0.1011
0.0175	0.1285	1.0278	1.0027	0.0053	0.1171
0.0114	0.0572	1.0278	1.0027	0.0025	0.0316
0.0317	0.2479	1.0278	1.0023	0.1284	0.2069
0.0038	0.1405	1.0278	1.0027	0.0836	0.1186
0.1154	0.2664	1.0278	1.0032	0.1662	0.3480

0.0284	0.1654	1.0278	1.0022	0.1367	0.3203
0.0225	0.1375	1.0278	1.0027	0.0385	0.1606
0.0209	0.1155	1.0278	1.0029	0.0552	0.1222
0.0223	0.1375	1.0278	1.0022	0.1345	0.1738
0.0124	0.1305	1.0278	1.0027	0.0027	0.0684
0.0345	0.1150	1.0278	1.0027	0.0055	0.0599
0.0182	0.1118	1.0278	1.0027	0.0055	0.1218
0.0038	0.0561	1.0278	1.0029	0.0007	0.0517
0.0031	0.1498	1.0278	1.0027	0.1726	0.2069
0.0146	0.1005	1.0278	1.0027	0.0055	0.0492
0.0327	0.1501	1.0278	1.0022	0.0084	0.1501
0.0345	0.0740	1.0278	1.0027	0.0055	0.0814
0.0126	0.1499	1.0278	1.0022	0.1269	0.1524
0.0327	0.1501	1.0278	1.0022	0.0035	0.1501
0.0200	0.1231	1.0278	1.0022	0.1269	0.1496
0.0180	0.1263	1.0278	1.0023	0.1284	0.2069
0.1170	0.2664	1.0278	1.0022	0.1367	0.3203
0.0088	0.1375	1.0278	1.0027	0.0385	0.1738
0.0130	0.1342	1.0278	1.0022	0.0836	0.0803
0.0183	0.1285	1.0278	1.0027	0.0033	0.1167
0.0366	0.1263	1.0278	1.0023	0.1284	0.2069

1.4.1 Gráfico de Pareto

```
[8]: # plot the results using matplotlib
import matplotlib.pyplot as plt

plt.scatter([solution.objectives[0] for solution in algorithm.result],
            [solution.objectives[1] for solution in algorithm.result])
#plt.xlim([600, 640])
#plt.ylim([0.10, 0.40])
plt.xlabel("$f_1(x)$ - Custo do Combustível$")
plt.ylabel("$f_2(x)$ - Emissão de SOx")
plt.show()
```

