Benchmark 2018: Operação eficiente de bombagem

Minimização dos custos de bombagem de uma estação elevatória

A. Andrade-Campos, J. Dias-de-Oliveira

Submetido: 24/10/2018

Resumo Um problema de minimização dos custos de bombagem de um subsistema de abastecimento de água é apresentado para que seja solucionado recorrendo a técnicas de otimização não-linear. Pretende-se encontrar a operação de bombagem mais eficiente respeitando os consumos de água necessários e os níveis limite do depósito. O desafio deste *benchmark* consiste em resolver este problema da forma mais eficiente e precisa do ponto de vista da otimização. Os resultados do problema devem incluir uma análise da evolução iterativa das variáveis de decisão, das funções objetivo e das restrições. Devem ser discutidas as variáveis de decisão e as técnicas de otimização utilizadas.

Palavras-Chave Eficiência de operações de bombagem · Minimização de custos · Otimização não-linear com restrições · *Benchmark*

1 Introdução

Hoje em dia, todos os sistemas de engenharia devem ser eficientes quer energeticamente quer economicamente. Os sistemas de bombagem, particularmente as estações elevatórias de sistemas de abastecimento de água, devem ser geridos de forma a reduzir os custos energéticos. Para isso, e com auxílio de depósitos, as bombas devem ser operadas nos ciclos/tarifários cujo custo energético é menor. Porém, a dimensão dos depósitos e a capacidade de bombagem não permitem que a totalidade de água necessária para os consumidores seja bombeada no periodo mais económico. Assim, é necessário um planeamento otimizado das operações de bombagem tendo em conta (i) as dimensões do depósito, (ii) as necessidades de água dos consumidores e (iii) as capacidades de bombagem. Este planeamento deve conduzir ao menor custo possível da energia de bombagem.

A. Andrade-Campos, J. Dias-de-Oliveira
Otimização não-linear em Engenharia
Ano letivo 2018/19
E-mail: gilac@ua.pt; jalex@ua.pt

2 Caso de estudo: subsistema F

Pretende-se minimizar o custo de bombagem de um subsistema de abastecimento que é composto por um depósito F, uma bomba P e os pontos de consumo VC e R, como mostra a figura 1.

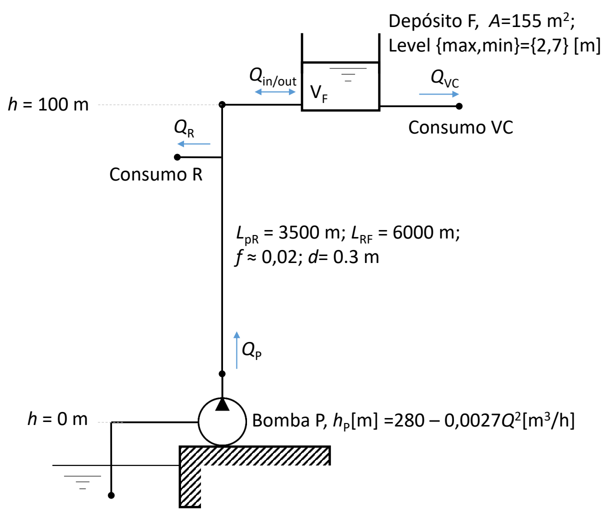

Figura 1. Subsistema de abastecimento de água. Pretende-se conhecer a operação de bombagem mais eficiente para o periodo de um dia.

O depósito F é um reservatório que está a uma cota h de 100 m, tem 155 m^2 de área A e uma altura de h = 9 m. Porém, por razões de segurança, este só

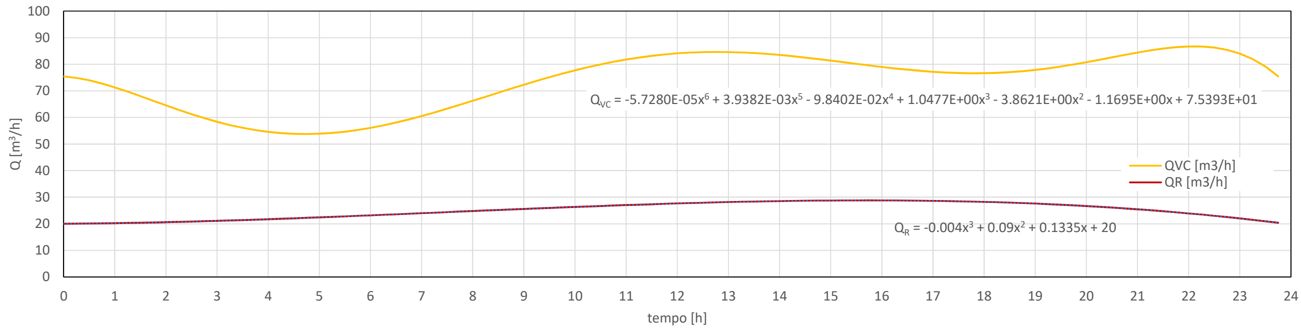

Figura 2. Consumos diários em m^3/h para as regiões VC e R. Os consumos são definidos pelos polinómios $Q_{\mathrm{VC}} = -5.7280 \times 10^{-5}t^6 + 3.9382 \times 10^{-3}t^5 - 9.8402 \times 10^{-2}t^4 + 1.0477t^3 - 3.8621t^2 - 1.1695t + 75.393$ e $Q_{\mathrm{R}} = -0.004t^3 + 0.09t^2 + 0.1335t + 20$ [h].

pode operar entre os níveis 2 e 7 m e, no início do dia, encontra-se com um nível de 4 m. O reservatório/depósito F abastece os consumidores da região VC. Também abastece os consumidores da região R quando a bomba não está em funcionamento, utilizando o ramal RF como retorno. A bomba, de 75% de eficiência e curva hidraulica $h_{\mathrm{P}}[\mathrm{m}] = 280 - 0{,}0027Q^2$, está à cota de 0 m e é responsável por enviar água para o reservatório F e para os consumidores R. O caudal está em m^3/h. As condutas são em ferro fundido, com um fator de atrito de Fanning igual a $f = 0{,}02$, têm o comprimento $L_{\mathrm{PR}} = 3{,}5$ e $L_{\mathrm{RF}} = 6$ km entre a bomba e o consumo R, e entre o consumo R e o reservatório F, respectivamente. Os consumos diários das regiões VC e R foram previstos e encontram-se representados na figura 2. O custo da energia (tarifário) varia ao longo do dia, como está listado na tabela 1.

Tabela 1. Tarifário energético ao longo do dia.

Intervalo [h]	Periodo horário	Custo [€/kWh]
[0, 2[	Vazio	0,0737
[2, 6[	Super vazio	0,06618
[6, 7[	Vazio	0,0737
[7, 9[	Cheia	0,10094
[9, 12[	Ponta	0,18581
[12, 24[	Cheia	0,10094

3 Formulação do sistema hidráulico

De acordo com a figura 1, as equações de balanço mássico (continuidade) no reservatório F e na conduta PR podem ser escritas respetivamente como

$$\frac{dV_{\mathrm{F}}}{dt} = Q_{\mathrm{in/out}} - Q_{\mathrm{VC}} \quad \text{e} \quad Q_{\mathrm{P}} = Q_{\mathrm{R}} + Q_{\mathrm{in/out}}. \tag{1}$$

Estas podem ser agregadas na seguinte equação:

$$A_{\mathrm{F}}\frac{dh_{\mathrm{F}}}{dt} = Q_{\mathrm{P}} - Q_{\mathrm{R}} - Q_{\mathrm{VC}}, \tag{2}$$

onde h_{F} e A_{F} são a altura e área do reservatório F, respetivamente. A condição de equilibrio energético na operação da bomba é dada por

$$h_{\mathrm{P}} - h_{\mathrm{perdas\ PR}} - h_{\mathrm{perdas\ RF}} = h_{\mathrm{F}} + h_{\mathrm{cota\ F}}, \tag{3}$$

onde $h_{\mathrm{cota\ F}} = 100$ m é a cota do depósito F. Sabendo que a bomba e as perdas de carga regem-se respetivamente pelas seguintes equações hidráulicas

$$h_{\mathrm{P}} = a_1 + a_2Q_{\mathrm{P}}^2 = 280 - 0{,}0027Q_{\mathrm{P}}^2, \tag{4}$$

$$h_{\mathrm{perdas}} = \frac{32fL}{d^5g\pi^2}Q^2, \tag{5}$$

então a equação toma a forma

$$aQ_{\mathrm{P}}^2 + bQ_{\mathrm{P}} + (c - h_{\mathrm{F}}) = 0, \tag{6}$$

com

$$a = a_2 - \gamma L_{\mathrm{PR}} - \gamma L_{\mathrm{RF}} \tag{7}$$

$$b = 2\gamma L_{\mathrm{RF}}Q_{\mathrm{R}} \tag{8}$$

$$c = a_1 - h_{\mathrm{cota\ F}} - \gamma L_{\mathrm{RF}}Q_{\mathrm{R}}^2 \tag{9}$$

$$\gamma = \frac{32f}{g\pi^2d^5}. \tag{10}$$

As equações 2 e 6 estão interligadas porque Q_{P} depende de h_{F} que, por sua vez, evolui com Q_{P}. A equação 6 é válida para qualquer instante t. Para cálculos médios num periodo de tempo Δt, Q_{R} e h_{F} podem ser substituidos pelos caudal médio $\bar{Q}_{\mathrm{R}}$ e nível médio $\bar{h}_{\mathrm{F}}$ desse periodo.

4 Formulação do problema de otimização

O objetivo é encontrar a operação da bomba que minimize o custo de bombagem. Para o caso de uma bomba hidráulica simples (sem variador de velocidade), a operação de bombagem resume-se a encontrar os periodos de atividade da bomba. Assim, as variáveis de decisão $\mathbf{x}$ podem tomar as seguintes configurações:

1. Variável contínua $\mathbf{x} \in [0,1]$ que representa a fração de tempo de funcionamento da bomba num periodo de tempo $\Delta t \in [0, 24]$ h;
2. Variável binária $\mathbf{x} \in \{0,1\}$ que representa o estado da bomba (*on/off*) num intervalo de tempo $\Delta t \in [0, 24]$ h.

O custo de operação da bomba é função do tarifário energético, da potência total $\dot{W}_\mathrm{P}$ e da eficiência η da bomba, e do estado ou tempo de operação da bomba. Este custo pode ser formulado para um dia de operação como

$$C(\mathbf{x}) = \int_0^{24\mathrm{h}} \frac{\dot{W}_\mathrm{P}(Q,h)}{\eta}\, x(t)\ \mathrm{Tarif}(t)\ \mathrm{d}t. \tag{11}$$

Sabendo que $\dot{W}_\mathrm{P} = \rho g Q_\mathrm{P} h_\mathrm{P}$ e que a bomba obdece à equação hidráulica $h_\mathrm{P} = a_1 + a_2 Q_\mathrm{P}^2$, o custo de operação pode ser escrito para uma soma discreta de $n_\mathrm{inc.t}$ periodos de tempo que perfazem as 24 h da seguinte forma:

$$C(\mathbf{x}) = \sum_{i=1}^{n_\mathrm{inc.t}} \frac{\rho g}{\eta} \left(a_1 \bar{Q}_{\mathrm{P},i} + a_2 \bar{Q}_{\mathrm{P},i}^3\right) x_i\ \mathrm{Tarif}_i, \tag{12}$$

onde $\bar{Q}_{\mathrm{P},i}$, Tarif_i e x_i são o caudal médio da bomba P, o custo da energia e o estado/tempo da bomba no periodo de tempo i, respetivamente. Note-se que a dependência do caudal $\bar{Q}_\mathrm{P}$ pela variável de decisão $\mathbf{x}$ pode ser desprezada[1].

Para além das restrições do universo das variáveis, as restrições do problema resumem-se à limitação das alturas do depósito:

$$2 \le h_\mathrm{F} \le 7. \tag{13}$$

Porém, estas restrições devem ser satisfeitas para todos os periodos de tempo ao longo do dia. Para um qualquer intervalo de tempo i, com $i = 1, \ldots, n_\mathrm{inc.t}$,

$$g_i = \begin{bmatrix} g_{1,i} \\ g_{2,i} \end{bmatrix} = \begin{bmatrix} 2 - h_{\mathrm{F},i} \\ h_{\mathrm{F},i} - 7 \end{bmatrix} \le \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{14}$$

Integrando a equação 2 para um periodo de tempo genérico i de duração Δt_i, a altura do depósito F no final do incremento é dada por

$$h_{\mathrm{F},i}(x_i) = h_{\mathrm{F},i-1}(x_{i-1}) + \frac{1}{A_\mathrm{F}} \left(x_i \bar{Q}_{\mathrm{P},i} \Delta t_i - \beta\right), \tag{15}$$

onde $\beta = \int_{t_{i-1}}^{t_i} Q_\mathrm{VC}\ dt + \int_{t_{i-1}}^{t_i} Q_\mathrm{R}\ dt$. Assim, o problema pode ser formulado classicamente como:

Procurar o estado/tempo $\mathbf{x}$ *de operação da bomba P de forma a*

$$\text{minimizar} \quad C(\mathbf{x}) \tag{16}$$
$$\text{sujeito a} \quad g_i \le 0, \quad i = 1, \ldots, n_\mathrm{inc.t} \tag{17}$$

5 Análise de sensibilidade

A sensibilidade das funções objetivo e das funções restrição podem ser calculadas por métodos aproximados como, por exemplo, o método das diferenças finitas. Porém, tendo em conta que as funções objetivo e de restrição podem ser escritas analiticamente, as sensibilidade podem ser encontradas através das suas derivadas. Assim, a derivada da função objetivo C para a variável de decisão α é dada por

$$\frac{\partial C}{\partial x_\alpha}(\mathbf{x}) = \frac{\rho g}{\eta} \left(a_1 \bar{Q}_{\mathrm{P},\alpha} + a_2 \bar{Q}_{\mathrm{P},\alpha}^3\right)\ \mathrm{Tarif}_\alpha. \tag{18}$$

Similarmente, as derivadas para as restrições de desigualdade de limite inferior e superior podem ser escritas de forma algébrica como[2]

$$\frac{\partial g_i}{\partial x_\alpha}(\mathbf{x}) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \text{para} \quad i < \alpha, \tag{21}$$

$$\frac{\partial g_i}{\partial x_\alpha}(\mathbf{x}) = \begin{bmatrix} -\bar{Q}_{\mathrm{P},\alpha} \Delta t_\alpha / A_\mathrm{F} \\ \bar{Q}_{\mathrm{P},\alpha} \Delta t_\alpha / A_\mathrm{F} \end{bmatrix} \text{para} \quad i \ge \alpha. \tag{22}$$

6 Avaliação numérica das funções objetivo/restrições - Programação

O seguinte código em python foi criado de forma a avaliar quer a função objetivo quer as restrições de forma automática. Este também avalia as sensibilidades pelas expressões na secção 5.

[1] $\bar{Q}_{\mathrm{P},i}$ não depende diretamente de x_i. Porém, pela equação 6, $\bar{Q}_{\mathrm{P},i}$ é função de $\bar{h}_\mathrm{F}$ e, por sua vez, $\bar{h}_\mathrm{F} = \frac{1}{2}(h_{\mathrm{F},i} + h_{\mathrm{F},i-1})$ é função de x_i, como indica a equação 15. Logo, o rigor matemático leva a $\bar{Q}_{\mathrm{P},i} = \bar{Q}_{\mathrm{P},i}(\bar{h}_\mathrm{F}(h_\mathrm{F}(x_i)))$.

[2] Caso a dependência do caudal com a variável $\mathbf{x}$ não seja desprezada, a equação 15 pode ser derivada implicitamente. Para isso, e pela equação 6,

$$\bar{Q}_{\mathrm{P},i} = \frac{-b - \sqrt{b^2 - 4a(c - \bar{h}_\mathrm{F})}}{2a} \tag{19}$$

será substituido na equação 15. A diferenciação implícita leva a que

$$\frac{\partial h_\mathrm{F}}{\partial x_i} = \frac{\Delta t\ \bar{Q}_{\mathrm{P},i}}{A_\mathrm{F} + \frac{1}{2} x_i \Delta t \left[b^2 - 4a(c - \bar{h}_\mathrm{F}\right]^{-\frac{1}{2}}}. \tag{20}$$

```
import numpy as np;
import math
import matplotlib.pyplot as plt

def Benchmark2018(nInc, x, iChart):
        # definição dos dicionários
        empty_timeIncrem = {
            'number': None,
            'startTime': None, 'duration': None,'endTime': None,
            'hFini': None,'hFfin': None,};
        fObjRest = {'fObj': None,'g1': [],'g2': []};
        Sensibil = {'dCdx': [],'dg1dx': [[0 for j in range(nInc)]for i in range(nInc)],
                        'dg2dx': [[0 for j in range(nInc)]for i in range(nInc)]};

        # Definição de funções do programa
        def Caudal_VC(ti, tf):
            # definição do polinómio
            a6 = -5.72800E-05; a5 = 3.9382E-03; a4=-9.8402E-02
            a3 = 1.0477; a2 = -3.8621; a1 = -1.1695; a0 = 7.53930E+01
            QVC = a6/7.*(tf**7.-ti**7.)+a5/6.*(tf**6.-ti**6.)+a4/5.*(tf**5.-ti**5.)
                + a3/4.*(tf**4.-ti**4.)+a2/3.*(tf**3.-ti**3.)
                + a1/2.*(tf**2.-ti**2.)+a0*(tf-ti)
            return QVC

        def Caudal_R(ti, tf):
            # definição do polinómio
            a3 = -0.004; a2 = 0.09; a1 = 0.1335; a0 = 20.0
            QR = a3/4.*(tf**4.-ti**4.)+a2/3.*(tf**3.-ti**3.)
                +a1/2.*(tf**2.-ti**2.)+a0*(tf-ti)
            return QR

        def tarifario(ti):
            # definição do tarifário usando o tempo inicial do incremento
            tarifHora = [None]*7; tarifCusto = [None]*7;
            set(tarifHora)
            tarifHora[0]= 0; tarifCusto[0]= 0.0737
            tarifHora[1]=2; tarifCusto[1]= 0.06618
            tarifHora[2]=6; tarifCusto[2]=  0.0737
            tarifHora[3]=7; tarifCusto[3]=  0.10094
            tarifHora[4]=9; tarifCusto[4]=  0.18581
            tarifHora[5]=12; tarifCusto[5]= 0.10094
            tarifHora[6]=24; tarifCusto[6]= 0.10094
            tarifF = 0.
            for i in range (0, 6):
                if (ti >= tarifHora[i]) & (ti < tarifHora[i+1]):
                    tarifF = tarifCusto[i]
                    break
            if tarifF == 0.: print("Erro no tarifário",ti,i); quit()
            return tarifF

        # Dados gerais
        g = 9.81; densidade = 1000.0
        hmin =  2.0; hmax = 7.0; hFixo = 100.0
        AF = 155.0; V0 = 620.0; hF0 = 4.; deltahF = 0.
        LPR = 3500; LRF = 6000
        f =  0.02; d =  0.3
        # Dados da bomba
        a1 = 280.; a2 = -0.0027; etaP = 0.75
        # variáveis constantes
        f32gpi2d5 = 32.0*f/(g*math.pi**2.0*d**5.)
        aRes = (a2*3600.**2.) - f32gpi2d5*LPR - f32gpi2d5*LRF

        # Inicialização dos vetores
        timeInc = []; CustoT = 0.;
        for i in range (0, nInc):
            # definição dos incrementos de tempo
            timeInc.append(empty_timeIncrem.copy());
            timeInc[i]['number'] = i + 1;
            if i==0:
                timeInc[i]['startTime'] = 0.
                hF = hF0; timeInc[i]['hFini']= hF
            else:
                timeInc[i]['startTime'] = timeInc[i-1]['endTime']
                timeInc[i]['hFini']= timeInc[i-1]['hFfin']
            timeInc[i]['duration'] =  24./nInc;
            timeInc[i]['endTime'] = timeInc[i]['startTime']+timeInc[i]['duration'];
            #
            # Cálculo dos volumes bombeados no incremento i
            QVC = Caudal_VC(timeInc[i]['startTime'], timeInc[i]['endTime'])
            QR = Caudal_R(timeInc[i]['startTime'], timeInc[i]['endTime']);
            QRmed= QR/timeInc[i]['duration']
```

```
            #
            # Ciclo iterativo de convergência (com tolerãncia=1.E-5)
            iter = 1; hFini = hF; hFmed = hF; deltahFold = 0.; tol = 1.E-6; maxIter = 8
            bRes = 2.*f32gpi2d5*LRF*QRmed/3600.
            while iter < maxIter:
                cRes = a1-hFixo -f32gpi2d5*LRF*(QRmed/3600)**2. - hFmed
                Qp = (-bRes -math.sqrt(bRes**2 - 4.*aRes*cRes))/(2.*aRes) *3600.
                deltahFn = (Qp*x[i]*timeInc[i]['duration']-QVC-QR)/AF
                hF = hFini + deltahFn; hFmed = hFini + deltahFn/2.;
                if math.fabs(deltahFn-deltahFold) > tol: deltahFold = deltahFn;
                else: break
                iter += 1
            timeInc[i]['hFfin']= hF
            #
            # Cálculo da energia utilizada
            WP = g*densidade/etaP*Qp/3600*(a1+a2*Qp**2.)    # in W
            tarifInc = tarifario(timeInc[i]['startTime'])*timeInc[i]['duration']/1000. # in Euro/W
            Custo =  x[i]*WP*tarifInc
            CustoT += Custo
            fObjRest['g1'].append(hmin - hF); fObjRest['g2'].append(hF - hmax);

            # Cálculo de sensibilidades (aproximadas, pois consideram que Qp é independente de x)
            Sensibil['dCdx'].append(WP*tarifInc);
            dgP=Qp*timeInc[i]['duration']/AF
            #dgP=Qp*timeInc[i]['duration']/(AF + 0.5*x[i]*timeInc[i]['duration']*(bRes**2-4.*aRes*cRes)**(-0.5))
            #
            # ciclo para cada dg1dx[i][j], onde i=alfa do x; j=inc. e j>=alfa
            for j in range (i, nInc): Sensibil['dg1dx'][i][j]=-dgP; Sensibil['dg2dx'][i][j]=dgP;
        # Guardar valores em Arrays
        fObjRest['fObj']=CustoT;

        # Construção da solução grafica
        if iChart == 1:
                x1=[];y1=[];z1=[]
                for i in range(0,nInc):
                        x1.insert(i,timeInc[i]['startTime']);
                        y1.insert(i,timeInc[i]['hFini']);
                        z1.insert(i,10*tarifario(i/(nInc/24)));
                plt.plot(x1,y1,x1,x[0:nInc],x1,z1);
                plt.plot(x1,y1);
                plt.title('Solução␣Proposta,␣Custo=%f' % CustoT); plt.xlabel('Tempo␣(h)');
                plt.ylabel('Nivel/␣status␣da␣bomba␣/␣Tarifario␣(x10)'); plt.grid();
                plt.show()

        #print ("end benchmark 2018")
        return fObjRest, Sensibil;

# main program (driver)
#-----------------------
nInc = 48 #96 #24
# Declaração de solução
x = [0 for i in range (0, nInc)];
for i in range (2, 8): x[i] = 1.;
for i in range (15, nInc): x[i] = 0.5;
x[20]=1.0;

fObjRest, Sensibil = Benchmark2018(nInc, x, 1)
print("CustoF=",fObjRest['fObj'], '\n')
```