



PIBIC/CNPq/UFPG-2010

**OTIMIZAÇÃO ("RETROFIT") DO PROJETO DE COLUNAS DE DESTILAÇÃO USANDO PROGRAMAÇÃO NÃO LINEAR INTEIRA MISTA ('MINLP').**

Rayla Pinto Vilar<sup>1</sup>, Antônio Carlos Brandão de Araújo<sup>2</sup>

**RESUMO**

Neste trabalho, uma abordagem algorítmica para a resolução de problemas de projeto ótimo de colunas de destilação será investigada. Primeiro, consideraremos o problema quando o número de estágios na coluna de destilação é conhecido e o que se pretende determinar são os melhores estágios para as alimentações. Em seguida, consideramos o problema de encontrar os melhores estágios para as alimentações e simultaneamente o número de estágios para uma determinada separação. No esquema adotado neste trabalho, as equações e restrições descrevendo a termodinâmica do sistema (equações que definem fugacidades, entalpias, etc) estão incluídos diretamente no problema de otimização. O sistema resultante apresenta uma grande dimensão e é extremamente disperso, o que conclama o uso de ferramentas computacionais que manipulem eficientemente técnicas de matriz tanto na programação não linear (NLP) quanto na programação inteira mista (MIP). A ferramenta de modelagem e simulação Matlab/Simulink e a ferramenta de otimização Tomlab serão utilizadas para a determinação do projeto otimizado de uma coluna de destilação industrial. Várias rotinas termodinâmicas serão implementadas em Matlab concomitante com o modelo para efeito comparativo entre os resultados das simulações.

**Palavras-chave:** Colunas de Destilação; Otimização; Programação Não linear inteira mista ("MINLP").

**Optimization ("retrofit") the design of distillation columns using non-linear programming whole mixed ('MINLP').**

**ABSTRACT**

In this paper, an algorithmic approach to solving problems of optimal design of distillation columns will be investigated. First, consider the problem when the number of stages in the distillation column is known and what one intends to stand is to determine the best placements for the feeds. Then we consider the problem of finding the best placements for the feeds and simultaneously the number of stages for a given separation. In the scheme adopted in this work, the equations and constraints describing the thermodynamics of the system (equations defining fugacities, enthalpies, etc.) are included directly in the optimization problem. The system features a large and extremely sparse, which urges the use of computational techniques for efficiently handling both the matrix nonlinear programming (NLP) and mixed integer programming (MIP). The modeling and simulation tool Matlab / Simulink and Tomlab optimization

<sup>1</sup> Aluna do curso de Engenharia Química, Unidade Acadêmica de Engenharia Química, UFPG, Campina Grande, PB, E-mail: raylavilar@gmail.com

<sup>2</sup> Engenheiro Químico, Professor Doutor, Unidade Acadêmica de Engenharia Química, UFPG, Campina Grande, PB, E-mail: antonio@deq.ufcg.edu.br

tool will be used for determining the optimum design of an industrial distillation column. Several thermodynamic routines will be implemented in tandem with Matlab model for comparison between the simulation results.

**Keywords:** Distillation Columns, Optimization, non-linear programming whole mixed ('MINLP').

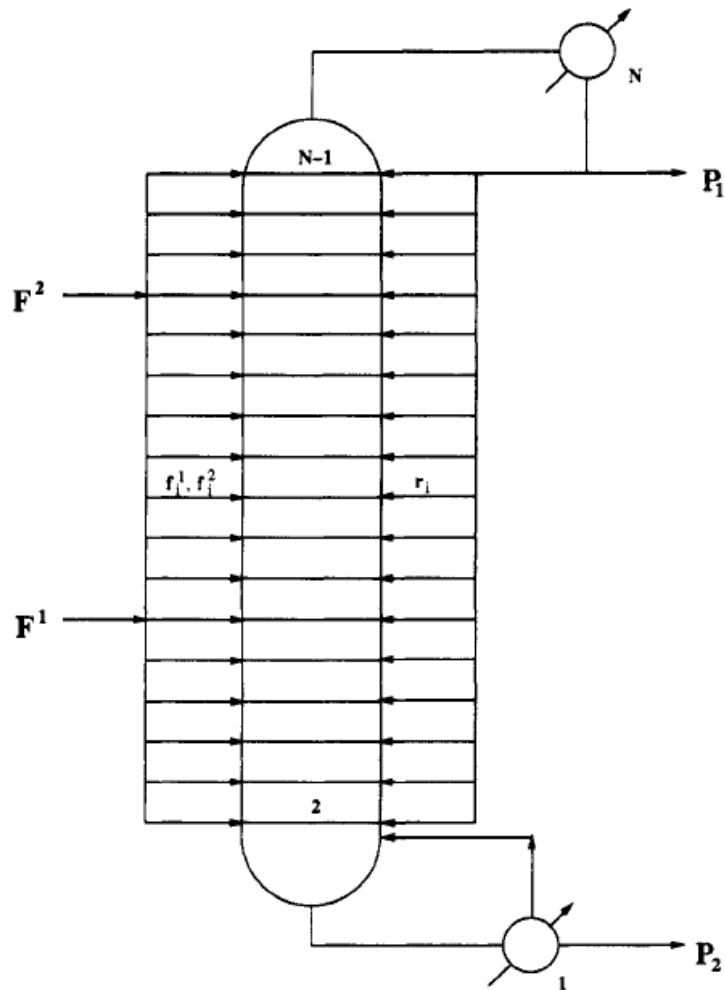
## INTRODUÇÃO

A destilação é o um dos mais usados métodos de separação na indústria química. Infelizmente, os custos envolvidos com este processo, especialmente sua operação, são muito elevados. Este problema faz da síntese de colunas de destilação uma tarefa muito importante, apesar das dificuldades relacionadas à complexidade do sistema e modelos de equilíbrio. Para a resolução destes problemas, a otimização matemática constitui uma ferramenta bastante adequada a fim de se encontrar a melhor solução para o problema em interesse.

Um dos primeiros esforços para resolver problemas usando um procedimento de otimização foi feito por Srygley e Holanda [Srygley e Holanda, 1965] utilizando o procedimento de busca desenvolvido por Hooke e Jeeves [Hooke e Jeeves, 1961]. O procedimento proposto foi limitado à determinação do número mínimo de estágios necessários para fornecer uma separação específica em uma determinada razão de refluxo. Sargent e Gaminibandara [Sargent e Gaminibandara, 1976] consideraram o problema mais geral da ótima configuração de colunas necessárias para efeito de uma determinada separação. Ricker e Grens [Ricker e Grens, 1974] propuseram um processo iterativo baseado na formulação de Naftali e Sandholm [Naftali e Sandholm, 1971] para a minimização do número de estágios necessários para realizar uma separação especificada para uma determinada razão de refluxo. Al-Haj-Ali e Holanda [Al-Haj-Ali e Holanda, 1979], determinaram as ótimas condições de funcionamento exigidas para uma coluna de destilação existentes assim como a concepção de condições para uma nova coluna. Fisher et al. [Fisher et al., 1985] propuseram uma generalização para a solução baseada na concepção e modelos econômicos simplificados com relativamente poucas composições de saída fixadas pelas condições de mercado e ambientais. Kumar e Lucia [Kumar e Lúcia, 1987] modificaram a programação quadrática sucessiva (SQP) de Wilson [Wilson, 1963], Han [Han, 1976], e Powell [Powell, 1978], de tal forma que a matriz de Hessian da função do Lagrangeano refletisse corretamente a termodinâmica dos processos de separação.

Muitos dos problemas envolvidos na operação de plantas, na sua concepção, localização e programação envolvem variáveis que não são contínuas, mas que podem assumir valores inteiros. Variáveis de decisão para as quais os níveis são uma dicotomia - instalar ou não uma nova peça de equipamento, por exemplo - são chamadas de "0-1" ou variáveis binárias. Outras variáveis inteiras podem ser números reais 0, 1, 2, 3, e assim por diant. Às vezes nós podemos tratar números inteiros como se fossem variáveis contínuas, especialmente quando o intervalo de uma variável contém um grande número de números inteiros, tais como 100 estágios em uma coluna de destilação, e depois arredondar a solução ideal para o valor inteiro mais próximo. Embora este procedimento leve a uma solução sub-ótima, a solução é bastante aceitáveis a partir de uma perspectiva prática. No entanto, para um pequeno intervalo de uma variável, como 1 a 3, quando a solução ideal produz um valor de 1,30 temos menos confiança nos arredondamentos.

Primeiro vamos classificar os tipos de problemas que são encontradas na otimização com variáveis discretas. O caso mais geral é a programação inteira mista (MIP) na qual a função objetivo depende de dois conjuntos de variáveis,  $x$  e  $y$ , onde  $x$  é um vetor de variáveis contínuas e  $y$  é um vetor de variáveis do tipo inteiro. Um problema envolvendo apenas variáveis inteiras é classificado como uma programação inteira (PI) pura. Finalmente, um caso especial da programação inteira pura é o de programação inteira binária (BIP), no qual todas as variáveis  $y$  são 0 ou 1. Muitos problemas MIP são lineares na função objetivo e restrições, e portanto estão sujeitos a solução de programação linear. Estes problemas são chamados de programação linear inteira mista (MILP). Problemas envolvendo variáveis discretas em que algumas das funções são não-lineares são chamados de programação não-linear inteira mista (MINLP). Em outras palavras, um método comum de síntese é a programação não-linear inteira mista (MINLP). MINLP oferece a possibilidade de executar o sistema de síntese e otimização simultaneamente. O método tem três etapas: (a) construir uma superestrutura, (b) MINLP gerar o modelo da superestrutura, e (c) encontrar a melhor estrutura e condições de operação, utilizando a ferramenta adequada. No caso de colunas de destilação a superestrutura pode ser definida conforme a Figura 1.



**Figura 1. Superestrutura para a determinação simultânea do número de estágios e localização dos estágios de alimentação.**

Um exemplo de modelo de superestrutura pode ser encontrado em Viswanathan e Grossmann [Viswanathan e Grossmann, 1993]. A solução pode ser encontrada através da aplicação de ferramentas de otimização para a solução de problemas de programação não-linear interia mista (MINLP).

## MATERIAL E MÉTODOS

Este trabalho foi desenvolvido no Laboratório de Experimentação Numérica de Processos do Departamento de Engenharia Química no Centro de Ciências e Tecnologia da Universidade Federal de Campina Grande – PB.

## ESTUDO DE CASO

O estudo de caso consiste em uma coluna destilação, cujas alimentações são uma mistura de etano, propano, n-butano, n-pentano e n-hexano. A coluna em estudo possui dezesseis estágios, sendo o primeiro representando pelo o condensador e último pelo refeedor. As alimentações estão localizadas nos estágios seis e nove. A alimentação que entra no estágio seis encontra-se a uma temperatura de 170F e a uma pressão de 300 psia. A alimentação que entra no estágio nove encontra-se a uma temperatura de 230F e uma pressão de 275 psia. O produto de topo é uma vapor que deixa a coluna a uma vazão de 15 lbmol/hr e o produto de fundo, um líquido destilado, que deixa a coluna a uma vazão de 5 lbmol/hr. O refluxo volta a coluna no estágio dois, a uma vazão de 150 lbmol/hr.

As especificações do problema que consistem das condições e das localizações dos estágios de todas as alimentações, pressão em cada estágio, taxas de fluxo total de todas as correntes, taxa de transferência de calor para ou a partir de todos os estágios com exceção do estágio um (condensador) e do estágio N

(refervedor), número total de estágio e a taxa de fluxo do vapor destilado, encontram-se representadas de forma mais detalhada na figura 2, a qual ilustra a coluna em estudo.

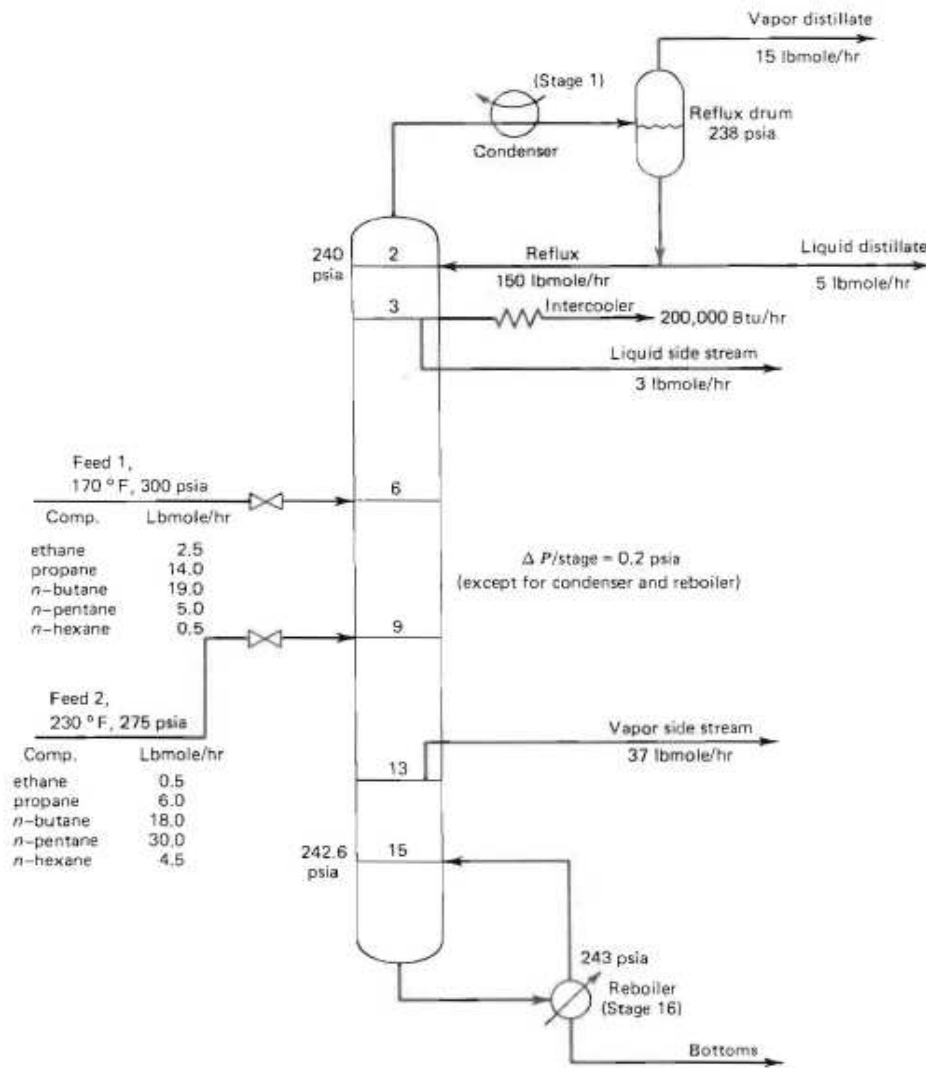


Figura 2 – Coluna de destilação utilizada como modelo para o problema.

## RESULTADOS E DISCUSSÃO

Os resultados da otimização inteira da coluna descrita na seção anterior mostram que os estágios de alimentação das duas correntes de entrada passam a ser 7 e 9, indicando que a configuração original é bem próxima da otimizada. No entanto, para alcançar a mínima fração molar de 0.9 de propano no destilado e 0.001 na corrente de fundo da coluna, e uma recuperação no destilado de  $0.05 \cdot \sum(F)$ , a carga térmica do reboiler passa a ser 2.21 GJ/h em comparação aos 1.65 GJ/h da configuração original. A tabela abaixo mostra os resultados dos perfis de composições, vazões e temperaturas.

Estagio	X <sub>ethane</sub>	X <sub>propane</sub>	X <sub>n-butane</sub>	X <sub>n-pentane</sub>	X <sub>n-hexane</sub>	Y <sub>ethane</sub>	Y <sub>propane</sub>	Y <sub>n-butane</sub>	Y <sub>n-pentane</sub>	Y <sub>n-hexane</sub>	L	V	T
1	0.076457	0.9	0.023477	6.61E-05	9.26E-09	0.175872	0.81544	0.008676	1.22E-05	5.80E-10	126.5212	6.620657	41.50057
2	0.033657	0.910998	0.055047	0.000298	1.18E-07	0.081318	0.895866	0.022753	6.34E-05	8.83E-09	126.4871	135.4099	47.1
3	0.01636	0.865499	0.116939	0.001201	1.31E-06	0.041329	0.90614	0.05225	0.00028	1.10E-07	121.8961	135.3757	51.33152
4	0.009469	0.762291	0.223884	0.004344	1.28E-05	0.025383	0.863583	0.109911	0.001122	1.22E-06	116.4834	132.1458	56.7984
5	0.006692	0.608499	0.371268	0.013439	0.000102	0.019434	0.76864	0.207906	0.004007	1.17E-05	110.6417	126.733	64.64921
6	0.005483	0.438036	0.521123	0.034714	0.000644	0.017375	0.628194	0.342023	0.012315	9.32E-05	106.3174	120.8913	74.45625
7	0.004877	0.292911	0.623493	0.075485	0.003233	0.016668	0.473451	0.477616	0.031678	0.000587	123.0625	116.567	84.55724
8	0.001989	0.186748	0.679049	0.125526	0.006688	0.007203	0.333993	0.596088	0.061224	0.001492	121.9005	114.7151	94.05673
9	0.001085	0.112446	0.660213	0.206937	0.019318	0.004127	0.220446	0.655448	0.114798	0.005182	152.3929	113.5531	103.1989
10	0.000363	0.069952	0.684388	0.225614	0.019683	0.001404	0.142238	0.717056	0.133538	0.005764	154.2092	117.2835	107.5344
11	0.000118	0.041393	0.682662	0.255317	0.020509	0.000464	0.086762	0.747491	0.15884	0.006443	154.9146	119.0998	111.1687
12	3.69E-05	0.02299	0.650881	0.303512	0.02258	0.000147	0.049735	0.744888	0.197641	0.007589	154.3215	119.8052	114.8232
13	1.03E-05	0.011523	0.582418	0.378052	0.027997	4.18E-05	0.025954	0.704056	0.259743	0.010205	152.3774	119.2121	119.3625
14	2.81E-06	0.005481	0.485634	0.468629	0.040253	1.17E-05	0.012961	0.627004	0.343839	0.016184	149.7503	134.051	124.9664
15	7.44E-07	0.002441	0.371196	0.560296	0.066067	3.18E-06	0.006105	0.517616	0.446363	0.029913	147.1984	131.4239	131.9612
16	1.87E-07	0.001	0.256285	0.628307	0.114408	8.23E-07	0.002646	0.387537	0.550624	0.059192	18.32639	128.872	140.2511

Os programas em Matlab usados para a obtenção dos resultados estão descritos abaixo.

```

% Main distillation
clear all; clc; format compact
run C:\tomlab\startup % Initialize Tomlab
rmpath('C:\tomlab\optim') % Allow use of Matlab's fmincon (not Tomlab's)

load nominal % x0 = [x y L V T Qcond Qreb]'; i = {ethanol, water, ethylene
glycol};
parameters_dist

load permut % Per
load scale % Sv, Sn

% Column pressure profile [bar]
P(1,1) = 1.01325;
P(2:par.N,1) = P(1) + PD/(par.N-1)*((2:par.N)-1);

% Column liquid side stream profile [kmole/h]
U = zeros(par.N,1);
U(1) = 85; % spec (distillate rate)

% Column vapor side stream profile [kmole/h]
W = zeros(par.N,1);

% Column stage heat transfer profile [kJ/h]
Q = zeros(par.N,1);

% Feed specifications
z(1,:) = [0.85 0.15 0.00];
z(2,:) = [0.00 0.00 1.00];
F = [100; 80]; % kmole/h
PF = [3; 3]*1.01325; % bar
TF = [40; 80]; % C
NF = [6 2];
m = eye(par.N);
m = m(:,NF);
if length(F)~=size(m,2)
    msgbox('Wrong number of feed stages!', 'Severe error', 'error');
    return
end

% Feed condition
for i = find(F~=0)'
    [HF(i),NoA,NoA,NoA] = feed_dist(F(i),z(i,:),TF(i),PF(i),par);

```

```

end

LoD      = zeros(par.N,1); LoD(1) = 0.5; % spec (L/D)
u        = [P U W Q m LoD];
d        = [F HF(:) z];

lb = zeros(length(x0),1); lb(end) = -1e20;
x0 = Per*x0; x0 = Sv\x0; lb = Per*lb;

addpath('C:\tomlab\optim')
Name      = 'Extractive Distillation';
A         = [];
b_L       = [];
b_U       = [];
x_0       = x0;
x_L       = lb;
x_U       = Inf*ones(length(x0),1);
c_L       = zeros(length(x0),1);
c_U       = zeros(length(x0),1);
fLowBnd   = [];
x_min     = [];
x_max     = [];
x_opt     = [];
f_opt     = [];

HessPattern = []; % parameter not used by SOL solvers
ConsPattern = []; % parameter not used by SOL solvers
pSepFunc    = [];

f = 'con_f';
g = 'con_g';%[];
H = 'con_H';%[];
c = 'con_c';
dc = [];
d2c = [];

Prob = conAssign(f, g, H, HessPattern, x_L, x_U, Name, x_0, ...
                pSepFunc, fLowBnd, ...
                A, b_L, b_U, c, dc, d2c, ConsPattern, c_L, c_U, ...
                x_min, x_max, f_opt, x_opt);
Prob.user.u = u;
Prob.user.d = d;
Prob.user.par = par;
Prob.user.Per = Per;
Prob.user.Sv = Sv;
Prob.user.Sn = Sn;
Prob.LargeScale = 1; % Using sparse version of the solver

% Result = tomRun('npsol',Prob,1);
Result = tomRun('snopt',Prob,1);
xr      = Per*(Sv*Result.x_k); yr = reshape(xr(1:end-2),par.N,2*par.nc+3);
function [HF,psi,x,y] = feed_dist(F,z,TF,PF)

parameters_dist
c = length(z); % c includes all species
options = optimset('Display','iter','Diagnostics','on');

% Bubble point calculation
xB = z(:); % Bubble point requirement (together with PF)
yB = xB; % Initial guess
TB = TF; % Initial guess
x0 = [yB; TB];

```

```

r = fsolve(@gB,x0,options,xB,PF,c);
yB = r(1:c);
TB = r(end);

% Dew point calculation
yD = z(:); % Dew point requirement (together with PF)
xD = yD; % Initial guess
TD = TF; % Initial guess
x0 = [xD; TD];
r = fsolve(@gD,x0,options,yD,PF,c);
xD = r(1:c);
TD = r(end);

if TF <= TB
    psi = 0;
    x = xB;
    y = yB;
    [NoA,NoA,NoA,NoA,NoA,HF,NoA,NoA] =
prsrk(x,PF*1e5,TF+273.15,pc,Tc,w,k,cpig,DHf,DGf,'L',Thermo);
elseif TF >= TD
    psi = 1;
    x = xD;
    y = yD;
    [NoA,NoA,NoA,NoA,NoA,HF,NoA,NoA] =
prsrk(y,PF*1e5,TF+273.15,pc,Tc,w,k,cpig,DHf,DGf,'V',Thermo);
else
    % Two-phase calculation
    x = z(:); % Initial guess
    y = x; % Initial guess
    V = 0.5*F; % Initial guess
    L = V; % Initial guess
    x0 = [x; y; V; L]; d = [F; TF; PF; z'];
    r = fsolve(@gF,x0,options,d,c);
    x = r(1:c); y = r(c+1:2*c);
    [NoA,NoA,NoA,NoA,NoA,HL,NoA,NoA] =
prsrk(x,PF*1e5,TF+273.15,pc,Tc,w,k,cpig,DHf,DGf,'L',Thermo);
    [NoA,NoA,NoA,NoA,NoA,HV,NoA,NoA] =
prsrk(y,PF*1e5,TF+273.15,pc,Tc,w,k,cpig,DHf,DGf,'V',Thermo);
    V = r(end-1); L = r(end);
    HF = 1/F*(HV*V + HL*L); psi = r(end-1)/F;
end

function ceq = gB(x0,x,PF,c)

parameters_dist
y = x0(1:c);
T = x0(end);

[NoA,NoA,phiL,NoA,NoA,NoA,NoA,NoA] =
prsrk(x,PF*1e5,T+273.15,pc,Tc,w,k,cpig,DHf,DGf,'L',Thermo);
[NoA,NoA,phiV,NoA,NoA,NoA,NoA,NoA] =
prsrk(y,PF*1e5,T+273.15,pc,Tc,w,k,cpig,DHf,DGf,'V',Thermo);
K = phiL./phiV;

ceq = [y - K.*x; sum(y) - 1];

function ceq = gD(x0,y,PF,c)

parameters_dist
x = x0(1:c);

```

```

T = x0(end);

[NoA,NoA,phiL,NoA,NoA,NoA,NoA,NoA] =
prsrk(x,PF*1e5,T+273.15,pc,Tc,w,k,cpig,DHf,DGf,'L',Thermo);
[NoA,NoA,phiV,NoA,NoA,NoA,NoA,NoA] =
prsrk(y,PF*1e5,T+273.15,pc,Tc,w,k,cpig,DHf,DGf,'V',Thermo);
K = phiL./phiV;

ceq = [y - K.*x; sum(x) - 1];

function ceq = gF(x0,d,c)

parameters_dist
x = x0(1:c);
y = x0(c+1:2*c);
V = x0(end-1);
L = x0(end);

F = d(1);
TF = d(2);
PF = d(3);
z = d(4:end);

[NoA,NoA,phiL,NoA,NoA,NoA,NoA,NoA] =
prsrk(x,PF*1e5,TF+273.15,pc,Tc,w,k,cpig,DHf,DGf,'L',Thermo);
[NoA,NoA,phiV,NoA,NoA,NoA,NoA,NoA] =
prsrk(y,PF*1e5,TF+273.15,pc,Tc,w,k,cpig,DHf,DGf,'V',Thermo);
K = phiL./phiV;

ceq = [y - K.*x; F*z - V*y - L*x; sum(x) - 1; sum(y) - 1];
function c = minlp_c(x0,Prob)

parameters_dist

u = Prob.user.u;
d = Prob.user.d;
aux = N*(2*C + 3);
xva = reshape(x0(1:aux),N,2*C+3);
% States
x = xva(:,0*C+1:1*C);
y = xva(:,1*C+1:2*C);
L = xva(:,2*C+1);
V = xva(:,2*C+2);
T = xva(:,2*C+3);

% Manipulations
P = u(:,1);
U = u(:,2);
W = u(:,3);
Q = u(:,4);

% Disturbances
F = d(:,1);
HF = d(:,2);
z = d(:,3:end);

% Additional states for optimization
U(1) = x0(aux+1);
Q(1) = x0(aux+2);
Q(3) = x0(aux+3);

```



```

Q(N) = x0(aux+4);
m     = reshape(x0(aux+5:end),N,length(F));

model_dist

c0 = [M(:); E(:); Sx(:); Sy(:); 1e-6*H(:)]; % Attempting to scale the
energy balances (H)
c1 = sum(m,1) - 1;

c = [c0; c1'];

```

```

function f = minlp_f(x0,Prob)

parameters_dist

u = Prob.user.u;
d = Prob.user.d;
aux = N*(2*C + 3);
xva = reshape(x0(1:aux),N,2*C+3);
% States
x = xva(:,0*C+1:1*C);
y = xva(:,1*C+1:2*C);
L = xva(:,2*C+1);
V = xva(:,2*C+2);
T = xva(:,2*C+3);

% Manipulations
P = u(:,1);
U = u(:,2);
W = u(:,3);
Q = u(:,4);

% Disturbances
F = d(:,1);
HF = d(:,2);
z = d(:,3:end);

% Additional states for optimization
U(1) = x0(aux+1);
Q(1) = x0(aux+2);
Q(3) = x0(aux+3);
Q(N) = x0(aux+4);
m     = reshape(x0(aux+5:end),N,length(F));

f = 11/2e6*(-Q(N)) + 0.1*V(1);

```

```

function g = minlp_g(x0,Prob)

parameters_dist

u = Prob.user.u;
d = Prob.user.d;
aux = N*(2*C + 3);
xva = reshape(x0(1:aux),N,2*C+3);
% States
x = xva(:,0*C+1:1*C);
y = xva(:,1*C+1:2*C);
L = xva(:,2*C+1);
V = xva(:,2*C+2);
T = xva(:,2*C+3);

% Manipulations
P = u(:,1);

```

```

U = u(:,2);
W = u(:,3);
Q = u(:,4);

% Disturbances
F = d(:,1);
HF = d(:,2);
z = d(:,3:end);

% Additional states for optimization
U(1) = x0(aux+1);
Q(1) = x0(aux+2);
Q(3) = x0(aux+3);
Q(N) = x0(aux+4);
m = reshape(x0(aux+5:end),N,length(F));

g = zeros(length(x0),1);
g(N*(2*C+1)+1) = 0.1; % Dround(f)/Dround(V(1))
g(aux+4) = -11/2e6; % Dround(f)/Dround(Q(N))

g = sparse(g);

```

```

function H = minlp_H(x0,Prob)

```

```

parameters_dist

```

```

u = Prob.user.u;
d = Prob.user.d;
aux = N*(2*C + 3);
xva = reshape(x0(1:aux),N,2*C+3);
% States
x = xva(:,0*C+1:1*C);
y = xva(:,1*C+1:2*C);
L = xva(:,2*C+1);
V = xva(:,2*C+2);
T = xva(:,2*C+3);

```

```

% Manipulations

```

```

P = u(:,1);
U = u(:,2);
W = u(:,3);
Q = u(:,4);

```

```

% Disturbances

```

```

F = d(:,1);
HF = d(:,2);
z = d(:,3:end);

```

```

% Additional states for optimization

```

```

U(1) = x0(aux+1);
Q(1) = x0(aux+2);
Q(3) = x0(aux+3);
Q(N) = x0(aux+4);
m = reshape(x0(aux+5:end),N,length(F));

```

```

H = zeros(length(x0));

```

```

H = sparse(H);

```

```

% Parameters

```

```

Thermo = 'PR'; % Thermodynamic package = {'PR', 'SRK', 'RK'}
DP = 0.2; % Pressure drop from stage 2 to N-1

```

```

% Physical parameters
pc      = [709.8 616.410385 550.70829 487.326798 436.56359]'; % psia
pc      = pc*1e5/14.5038; % Pa
Tc      = [89.906 206.006002 305.618002 385.412001 453.47]'; % F
Tc      = (Tc - 32)/1.8 + 273.15; % K
w       = [0.1064 0.152 0.201 0.254 0.301]';
k(1,1) = 0;
k(1,2) = 1.1e-3;
k(1,3) = 9.6e-3;
k(1,4) = 7.8e-3;
k(1,5) = -0.01;
k(2,2) = 0;
k(2,3) = 3.3e-3;
k(2,4) = 0.0267;
k(2,5) = 7.0e-4;
k(3,3) = 0;
k(3,4) = 0.0174;
k(3,5) = -5.6e-3;
k(4,4) = 0;
k(4,5) = 0;
k(5,5) = 0;
k       = k + k';
cpig    = [40.33 51.92 71.34 88.05 104.40; 134.22 192.45 243.00 301.10
352.30; 1655.50 1626.50 1630.00 1650.20 1694.60;
73.22 116.80 150.33 189.20 236.90; 752.87 723.60 730.42 747.60
761.60]'; % J/mol-K
DHF     = [-8.3820e+4 -1.0468e+5 -1.2579e+5 -1.4676e+5 -1.6694e+5]'; %
J/mol
DGF     = [-3.1920e+4 -2.4290e+4 -1.6570e+4 -0.8650e+4 +0.0150e+4]'; %
J/mol

% Number of stages and number of components
N = 16;
C = length(DGF);

% EoS computation and property calculations
% The (generalized) equation of state is of the form:
%  $p = RT/(V - b) - a(T)/[(V + \text{epsi}*b)*(V + \text{sigm}*b)]$ 

% Inputs:
% z      = number of moles of all species (n x 1 column vector) [mol]
% p      = pressure [Pa]
% T      = temperature [K]
% pc     = critical pressure of all components (n x 1 column vector)[Pa]
% Tc     = critical temperature of all components (n x 1 column vector)[K]
% w      = acentric factor of all components (n x 1 column vector)
% k      = binary parameters (n x n symmetric matrix)
% cpig   = ideal gas heat capacity coefficients (n x m symmetric matrix)
[J/mol-K]
% DHf    = standard enthalpy of formation for ideal gas (298.15 K and 1 atm)
(n x 1 column vector) [J/mol]
% DGF    = standard Gibbs energy of formation for ideal gas (298.15 K and 1
atm) (n x 1 column vector) [J/mol]
% ph     = phase ('L' or 'V')
% flag   = chosen EoS ('PR', 'SRK', or 'RK')
% Outputs:
% V      = molar volume [m3/mol]
% Z      = compressibility factor
% phi    = fugacity coefficient
% A      = Helmholtz energy [J/mol]
% S      = entropy [J/(mol K)]
% H      = enthalpy [J/mol]
% U      = internal eenergy [J/mol]
% G      = Gibbs energy [J/mol]

```

```

function [rho,Z,phi,A,S,H,U,G] =
prsrk(z,p,T,pc,Tc,w,k,cpig,DHf,DGf,ph,flag)

z = z(:); % z is always a column vector
R = 8.314; % m3 Pa/(mol K) = J/mol-K
Tref = 298.15; % K (reference temperature)
Pref = 1; % bar (reference pressure)
DSf = 1/T*(DHf - DGf); % standard entropy of formation [J/(mol K)].
Calculated at the given temperature, not the reference temperature.

switch flag
case {'PR'}
    epsi = 1 - sqrt(2);
    sigm = 1 + sqrt(2);
    omeg = 0.07780;
    psi = 0.45724;
    mi = [0.37464 1.54226 0.26992];
    m = [ones(length(w),1) w -w.^2]*mi';
case {'SRK'}
    epsi = 0;
    sigm = 1;
    omeg = 0.08664;
    psi = 0.42748;
    mi = [0.480 1.574 0.176];
    m = [ones(length(w),1) w -w.^2]*mi';
case {'RK'}
    epsi = 0;
    sigm = 1;
    omeg = 0.08664;
    psi = 0.42748;
    m = ((T./Tc).^(-1/4) - 1)./(1 - (T./Tc).^0.5);
end

alfa = (1 + m.*(1 - (T./Tc).^0.5)).^2;
ai = psi*(R^2)*(Tc.^2)./pc.*alfa;
bi = omeg*R*Tc./pc;
Q = ((ai*ai').^0.5).*(1 - k);
a = z'*Q*z;
b = z'*bi;
dQdT = psi*(R^2)*(k - 1).*(Tc*Tc')./((pc*pc').^0.5).*(1/(2*T^0.5)).*...
((m./(Tc.^0.5))*(alfa.^0.5)' + (alfa.^0.5)*(m./(Tc.^0.5))');
dadT = z'*dQdT*z;

% Coefficients of the EoS model equation
c(1) = 1;
c(2) = (epsi + sigm - 1)*b - R*T/p;
c(3) = (epsi*sigm - (epsi + sigm))*b^2 - R*T*(epsi + sigm)/p*b + a/p;
c(4) = -epsi*sigm*b^3 - R*T*epsi*sigm/p*b^2 - a*b/p;

% Roots
r = roots(c); %ir = imag(r); r(ir<1e-8) = real(r(ir<1e-10));
for i = 1:length(r), index(i) = isreal(r(i)); end
r = r(index);
if upper(ph) == 'L'
    V = min(r);
else
    V = max(r);
end
rho = 1/V;
Z = p*V/(R*T);

```

```

u = epsi + sigm;
v = epsi*sigm;
Bs = b*p/(R*T);
AR = a/(b*(sqrt(u^2 - 4*v)))*log((2*Z + Bs*(u - sqrt(u^2 - 4*v)))/(2*Z +
Bs*(u + sqrt(u^2 - 4*v)))) - ...
R*T*log((Z - Bs)/Z) - R*T*log(V/(R*T/p));
SR = R*log((Z - Bs)/Z) + R*log(V/(R*T/p)) - 1/(b*(sqrt(u^2 -
4*v)))*dadT*...
log((2*Z + Bs*(u - sqrt(u^2 - 4*v)))/(2*Z + Bs*(u + sqrt(u^2 -
4*v))));
HR = AR + T*SR + R*T*(Z - 1);

% From Aly, F. A. and Lee, L. L., "Self-Consistent Equations for
% Calculating the Ideal Gas Heat Capacity, Enthalpy, and Entropy",
% Fluid Phase Equilibria, Vol. 6, 1981, p. 169, we have:
% Cp = C1 + C2*((C3/T)/sinh(C3/T))^2 + C4*((C5/T)/cosh(C5/T))^2.
% But:
% Hig = z'*(DHf + int(Cp,T,Tref,T)).
% Sig = z'*(DSf + int(Cp/T,T,Tref,T) - R*ln(P/Pref)) - R*z'*ln(z).
Hig = z'*(DHf + cpig(:,1)*(T - Tref) +
2*cpig(:,2).*cpig(:,3).*(1./(exp(2*cpig(:,3)/T) - 1) -
1./(exp(2*cpig(:,3)/Tref) - 1)) ...
+ 2*cpig(:,4).*cpig(:,5).*(1./(exp(2*cpig(:,5)/T) + 1) -
1./(exp(2*cpig(:,5)/Tref) + 1)));
Sig = z'*(DSf + cpig(:,1)*log(T/Tref) -
cpig(:,2).*log((exp(2*cpig(:,3)/T) - 1)./(exp(2*cpig(:,3)/Tref) - 1))) ...
+ cpig(:,4).*log((exp(2*cpig(:,5)/T) + 1)./(exp(2*cpig(:,5)/Tref) +
1))) + 2*(cpig(:,2).*cpig(:,3) - cpig(:,4).*cpig(:,5))*(1/T - 1/Tref) ...
+ 2*cpig(:,2).*cpig(:,3).*(1./(T*(exp(2*cpig(:,3)/T) - 1)) -
1./(Tref*(exp(2*cpig(:,3)/Tref) - 1))) ...
+ 2*cpig(:,4).*cpig(:,5).*(1./(T*(exp(2*cpig(:,5)/T) - 1)) -
1./(Tref*(exp(2*cpig(:,5)/Tref) - 1))) ...
- R*log(p/Pref)) - R*z'*log(z);
abar = (2*z'*Q - a*ones(1,length(z)))'; bbar = bi;
phi = exp((Z - 1)*bbar/b - log((V - b)*Z/V) + (a/(b*R*T))/(epsi -
sigm)*log((V + sigm*b)/(V + epsi*b))*...
(1 + abar/a - bbar/b));

H = Hig + HR;
S = Sig + SR;
U = H - p*V;
A = U - T*S;
G = H - T*S;

```

## CONCLUSÃO

Este trabalho de iniciação científica detalhou um procedimento para otimização ("retrofit") do projeto de colunas de destilação usando programação não-linear inteira mista. Os resultados sugerem que o procedimento, usando o software Matlab com pacote de propriedades termodinâmicas rigoroso, é bastante eficiente e pode ser usado para o projeto de colunas de destilação mais complexas.

## AGRADECIMENTOS

Ao CNPq pela bolsa de Iniciação Científica e ao professor orientador.

## REFERÊNCIAS BIBLIOGRÁFICAS

- Srygley, J. M.; Holland C. D., AIChE J 11 (1965) pp. 695-701.
- Hook, R.; Jeeves T. A., J. Assoc. Comp. Math. 8 (1961) pp. 212-229.
- Sargent, R. W. H.; Gaminibandara K., Optimum Design of Plate Distillation Columns. Optimization in Action, (L. D. W. Dixon, Ed.) Academic Press, New York 1976.
- Ricker, N.L.; Grens E. A., AIChE J. 20 (1974) pp. 238-244.
- Naphtali, L.M.; Sandholm. AIChE J. 17 (1971) pp. 148-153.
- Al-Haj-Ali, N.S.; Holland C. D., Hydrocarbon Processing, July (1979) pp. 165-175.
- Fisher, W.R.; Doherty M. F.; Douglas J. M., AIChE J. 31 (1985) pp. 1538-1547.
- Kumar, A.; Lucia A., IMACS J. Appl. Numer. Math. 3 (1987) pp. 409-425.
- Wilson, R.B., Ph.D. Thesis, Harvard University, 1963.
- Han S.P., Math. Program 11 (1976) pp.263-282.
- Powell, M. J. D., A Fast Algorithm for Nonlinearly Constrained Optimization Problems, in: Lecture Notes in Mathematics (G. A. Watson, Ed.), Springer-Verlag, Berlin 1978.
- Viswanathan J. and Grossmann, I. E., Optimal feed locations and number of trays for distillation columns with multiple feeds. Ind. Eng. Chem. Res., 1993, 32 (11), 2942-2949.
- Mathwork, INC. MATLAB: High-performance numeric computations and visualization software. MATLAB user's guide, 1997.