Synthesis and Analysis of Separation Processes for Extracellular Chemicals Generated from Microbial Conversions

Wenzhao Wu[1†], Kirti Yenkie[1,2†*], and Christos T. Maravelias[1]

[†] Equal Contributors

[1] Dept. of Chemical and Biological Engineering and DOE Great Lakes Bioenergy Research Center, University of Wisconsin-Madison, Madison, WI 53706

[2] Department of Chemical Engineering, Henry M. Rowan College of Engineering, Rowan University, Glassboro, NJ 08028

ADDITIONAL FILE

A. Materials for EX NSL products

A.1. Separation scheme for EX NSL products

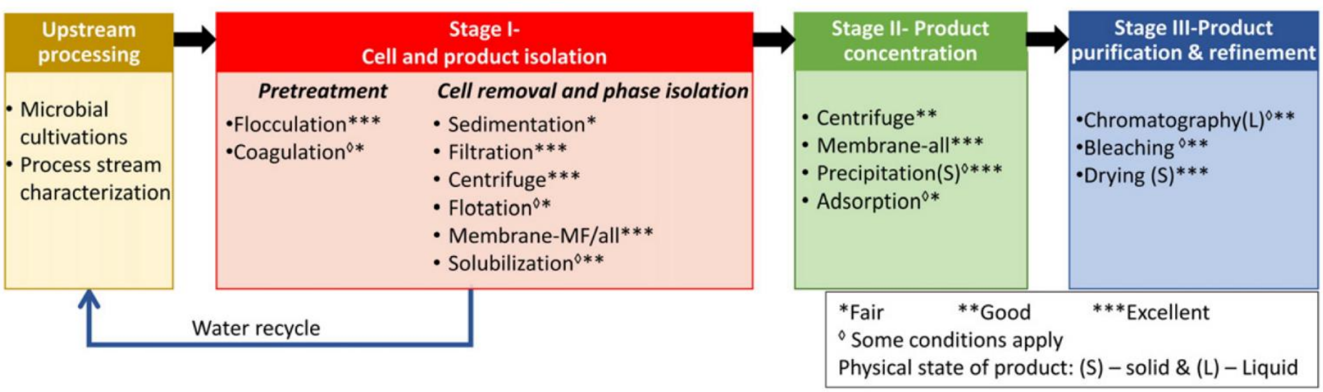

Figure A1. Separation scheme for extracellular (EX) and insoluble (NSL) product.

The technology suitability has been divided into grades - fair (*), good (**) and excellent (***), if applicable under certain conditions (◊) and physical state of the product (solid (S) and liquid (L)). In stage I, the option Membrane - MF/all denotes MF to be suitable for cell removal and all membrane options (MF, UF, NF, RO) suitable for phase isolation

A.2. Cost-titer curve for EX NSL HV SLD CMD product without color imparting impurities

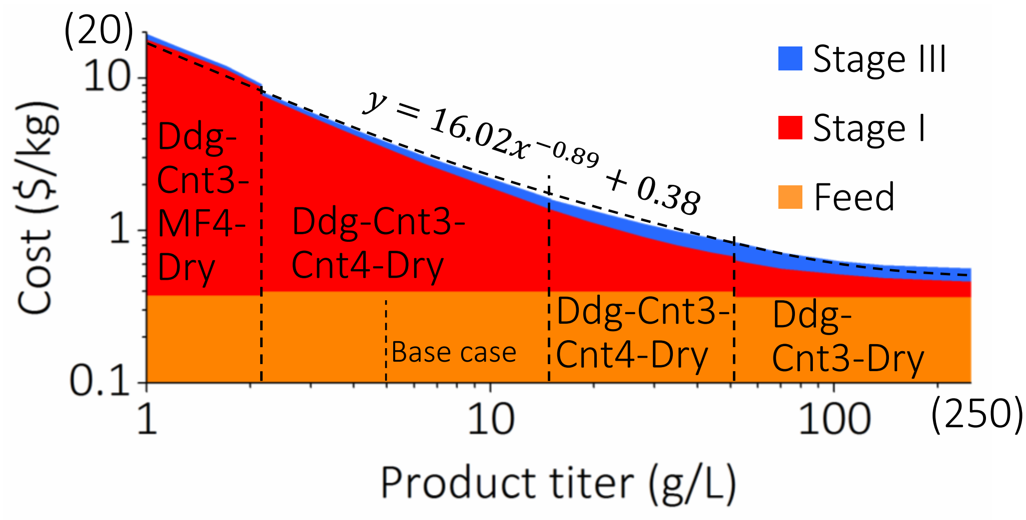

Figure A2. Cost-titer curve for EX NSL HV SLD CMD product, where no color imparting impurities are assumed to exist, and thus no Blc is necessary.

A.3. Extension to EX NSL HV LQD CMD product

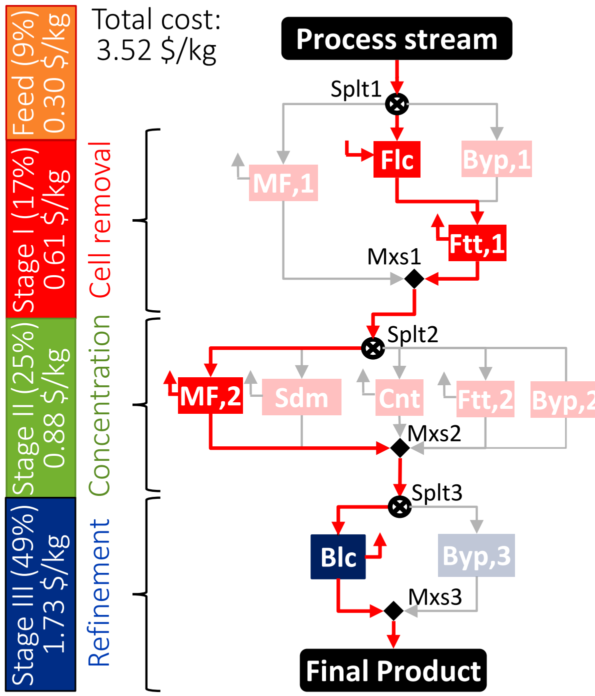

Figure A3. Superstructure (including all the units and streams) and optimal solution (the highlighted parts) for EX NSL HV LQD CMD product. The active streams are shown by bold red lines and selected technologies are highlighted in different colors corresponding to each stage: red for stage I, green for stage II, and blue for stage III. Cost distribution is shown by the numbers on the left bar.

B. Materials for EX SOL products

B.1. Separation scheme for EX SOL products

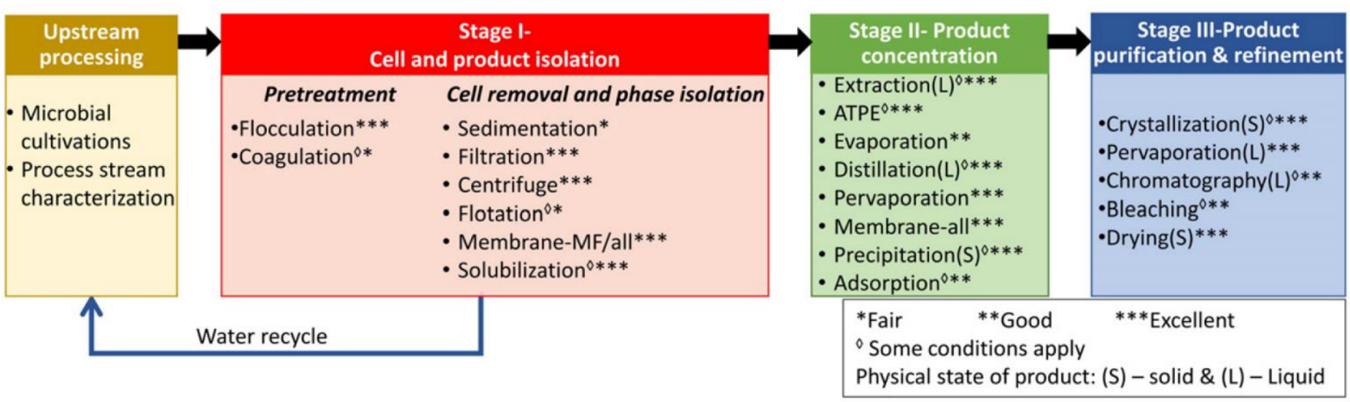

Figure B1. Separation scheme for extracellular (EX) and soluble (SOL) product.

The technology suitability has been divided into grades - fair (*), good (**) and excellent (***), if applicable under certain conditions (◊) and physical state of the product (solid (S) and liquid (L))

B.2. Variation in Ext partition coefficient

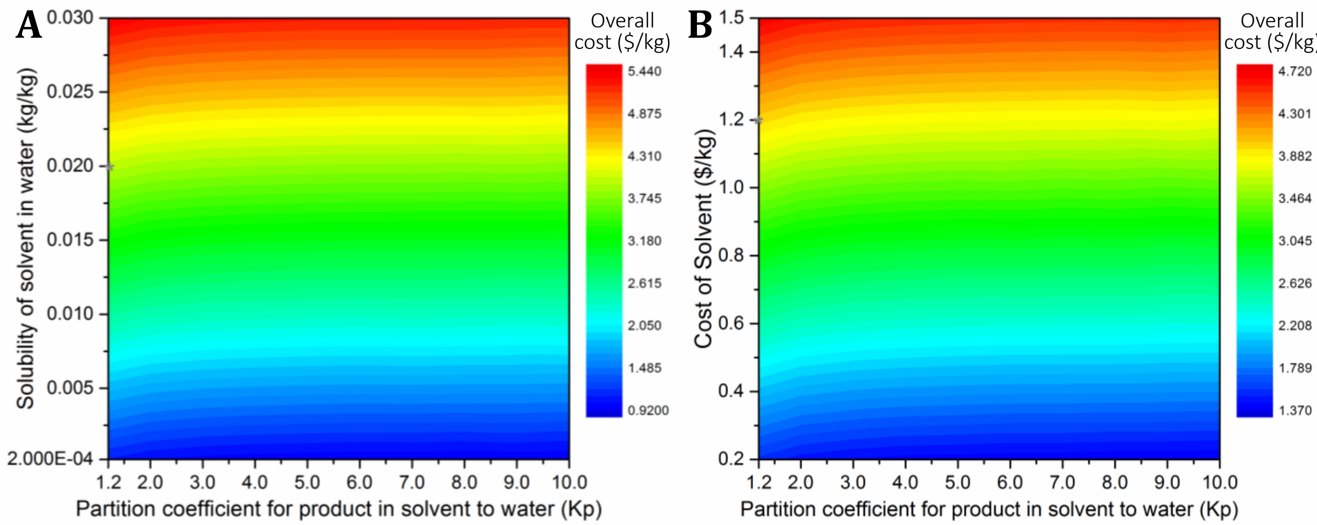

Figure B2. Variation in (A) partition coefficient versus solubility in water, and (B) partition coefficient versus cost of solvent. The base case values are represented by the grey asterisks.

C. Model parameters and input data for base cases:

Table B1. Important input parameters and product specifications

Parameter	Nominal value	Units
Product titer	5	g/L (kg/m3)
Cell titer	1.7	g/L (kg/m3)
Liquid co-product titer	0.85	g/L (kg/m3)
Solid co-product titer	0.85	g/L (kg/m3)
Desired production capacity	1000	kg/h
Annual operation time	330	days/year
Final product purity	95	wt% purity

Table B2. Particle size and density information

Component	Particle size (m)	Density (kg/m3)	Molecular weight (kg/mol)
Product	5E-6	1800	775
Cell	8E-6	1250	24.6
Liquid co-product	3.8E-10	950	85
Solid co-product	5E-7	1100	140
Water	3E-10	1000	18

Table B3. Utility and labor costs: (SuperPro Designer v8.5)

Utility	Cost per unit ($/unit)
Electricity	0.1 $/KWH
Cooling water	5E-5 $/kg
Steam	0.012 $/kg
Labor	20 $/laborer-h
Latent heat (steam)	2155.68 KJ/kg

Flocculation (Flc):
Flocculent added – 0.04 kg/m3
Residence time – 0.5 hr

Floc diameter – 5E-4 m
Flocculent cost – 5 $/kg

Sedimentation tank (Sdm):
Efficiency – 70%
Maximum concentrating factor – 20
Depth – 3m
Residence time – 6 hr

Centrifuge (Cnt):
Efficiency – 80%
Maximum concentrating factor – 30
Rotation speed – 9000 rpm

Filtration (Ftt):
Flux - 0.2 $m^3m^{-2}h^{-1}$
Retention factors (Ftt): 80%
Filter cost – 100 $/$m^2$
Replacement time – 2000 h

Microfiltration (MF):
Flux – 0.0856 $m^3m^{-2}h^{-1}$
Retention factor: less filtered components (liq co-product, water, solvents) – 0.15, filtered components (product, cell, solid co-product, salt, polymer) – 0.85,
Microfilter membrane cost – 500 $/$m^2$
Replacement time – 2000 h

Differential digestion (Ddg):
Agent required – 0.5 kg/kg NPCM (non-product solid materials – cells, solid co-product)
Cost of agent – 1.5 $/kg
Density of agent – 1400 kg/m^3

Solubilization (Slb):
Solvent required – 0.5 kg/kg product
Cost of solvent – 1.5 $/kg
Density of solvent – 1300 kg/m^3

Precipitation (Prc)
Efficiency of product precipitation – 98%
Anti-solvent required – 2 kg/kg product
Cost of anti-solvent – 1.8 $/kg
Density of anti-solvent – 925 kg/m^3

Distillation (Dst1):
Relative volatility: product – 1, water – 1.3, soluble co-product – 1.5
Heat of vaporization (KJ/kg): product – 573, water – 2257, soluble co-product – 1275
Feed quality, q_f = 1 (saturated liquid)
Vapor velocity – 3 m/s
Stage efficiency – 80%
Height of stage – 0.6 m
Reflux ratio multiplying factor – 1.3

ATPE (Atpe)
Partition coefficient in top phase: product – 4, water – 1, soluble co-product – 2, heavy solid – 0.001
Solubility of polymer in bottom phase – 0.005 (kg/kg)
Solubility of salt in top phase – 0.005 (kg/kg)

Polymer: Mol. Wt. – 450, Density – 1850 (kg/m^3), Cost – 2 $/kg
Salt: Mol. Wt. – 136, Density – 1636 (kg/m^3), Cost – 0.6 $/kg

Extraction (Ext):
Partition coefficient in solvent phase: product – 1.2, soluble co-product – 0.3, heavy solid– 0.0001
Solubility of solvent in water – 0.02 (kg/kg)
Solubility of water in solvent – 0.02 (kg/kg)
Solvent: Mol. Wt. – 78, Density – 810 (kg/m^3), Cost – 1.2 $/kg

Distillation (Dst2):
Relative volatility: solvent – 5.2, product – 1, water – 1.4, soluble co-product – 1.5
Heat of vaporization (KJ/kg): solvent – 592, product – 2000, water – 573, soluble co-product – 1275, heavy solid– 2257
Other parameters are the same with those in Dst1

Drying (Dry):
Sublimation of solvents – 97%
Heat of sublimation – 5000 KJ/kg
Ambient temperature: 20 ^{0}C
Freezing temperature: (-1) ^{0} C
Refrigerant inlet temperature – (-10) ^{0} C
Refrigerant outlet temperature – 0 ^{0} C
Specific heat (KJ/kg-^{0}C): Refrigerant – 12, water – 4.2, liquids (co product and added agents) – 1, product – 6.2, solid coproduct – 5.7, cell – 4.2
Heat transfer coefficient – 180 KJ/m^2-K-h

Chromatography (Chr):
Space time – 0.5 h
Column capacity – 95%
Width of chromatogram – 0.05 m
HETP – 0.0035
Ratio length to diameter – 0.14

Pervaporation (Pvp):
Flux – 0.055 $m^3m^{-2}h^{-1}$
Retention factor: product – 0.0002, water – 0.95, soluble co-product – 0.92, heavy solid – 0.99, salt – 0.99, polymer – 0.99, solvent – 0.001
Membrane cost – 1000 $/$m^2$
Replacement time – 2000 h

Bleaching (Blc):
Bleaching efficiency – 99%
Cost of GAC (bleaching agent) – 4$/kg
Replacement time – 360 h (15 days)

Other parameters:
Capital charge factor – 0.11
Base module cost multiplier – 5.4
Annual operation time – 330 days
Cost of feed – 0.25 $/kg (per kg product basis)

Table B4. Standard capacity, costs, scaling factors, labor requirements for technologies:

Unit operation (costing capacity)	Standard capacity (units)	Base costs (million $)	Scaling exponent (n)	Labors required (#/h)	Power required (KW/h)
Differential digestion (Flowrate)	40 m^3/h	0.474	0.5	1	0.1
Solubilization (Flowrate)	40 m^3/h	0.474	0.5	1	0.1
Flocculation (Volume)	2000 m^3	0.54	0.5	0.1	0.0002
Sedimentation (Area)	2500 m^2	1.13	0.57	0.1	0
Centrifuge (Sigma factor)	60000 m^2	0.275	0.65	1	12.79
Filtration (Area)	80 m^2	0.04	0.55	0.5	0.1
Microfiltration (Area)	80 m^2	0.75	0.55	1	0.1
Precipitation (Flowrate)	40 m^3/h	0.47	0.5	1	0.1
Freeze drying (Capacity)	600 kg/h	0.11	0.67	0.5	0.3
Distillation (Volume)	22.58 m^3	0.082	2.8	1	0
ATPE (Flowrate)	185 m^3/h	0.362	0.67	1	0.5
Extraction (Flowrate)	185 m^3/h	0.362	0.67	1	0.5
Chromatography (Volume)	0.633 m^3	0.775	0.67	1	0.33
Pervaporation (Area)	80 m^2	0.261	0.55	1	0.33
Bleaching (Volume)	0.27 m^3	0.1	0.67	1	0.33

D. Sample Case Study Model Equations (EX NSL HV SLD CMD products):

Notations

- *The 'uppercase italic Latin fonts (not colored)' are for variables (optimization variables)*
- *The uppercase Latin fonts and lowercase Greek fonts in red are the specified input parameters*
- *The uppercase Latin fonts and lowercase Greek fonts in green color are for the parameters evaluated from inputs available*
- *The parameter or variable to be evaluated is always on the L.H.S. of the equation*

C.1 Indices and sets

$i \in \boldsymbol{I}$ - technologies (used as subscript to variables)

{*flc* - flocculation,

sdm - sedimentation,

splt# and *mxs#* - splitters and mixers,

cnt# - centrifugation and # = {1, 2, 3, 4},

ftt# - filtration and # = {1, 2,3,4},

mbr - membranes, MF (*mf#*) – microfiltration {1, 2, 3, 4}

byp - bypass

ddg - differential digestion of NPCM (non-product cell materials)

htt – heat pretreatment

slb - solubilization of product

prc - precipitation

dry - drying}

$j \in \boldsymbol{J}$ - stream (used as subscript to variables)

{1, 2, 3, 4, ………………….., 101}

$k \in \boldsymbol{K}$ - components (used as subscript to variables)

{Cell – biomass, W – water, Prd (HLA) – product, Lcoprd – Light coproducts, Scoprd- soluble coproducts, Flcnt – flocculent added, Solv – solvent, DAgnt – digestion agent, AnSol – antisolvent in precipitation}

$s \in \boldsymbol{NS}$ – stages {*s1, s2, s3*}

$l \in \boldsymbol{L}$ - utilities {electricity (*elec*), steam (*stm*), cooling water (*cwt*), refrigerant (*rfrg*)}

C.2 Subsets

Subsets for technologies

$\boldsymbol{I}^{\mathbf{CST}}$ – technologies with costs

{*htt,slb,ddg,mf1,flc,sdm1,cnt1,ftt1,mf2,sdm2,cnt2,ftt2,mf3,sdm3,cnt3,ftt3,mf4,sdm4,cnt4,ftt4,prc,dst,mf5,sdm5,cnt5,ftt5,blc,dry*}

$\boldsymbol{I}^{\mathbf{CF}}$ – technologies with concentration factor

{*mf1,sdm1,cnt1,ftt1,mf2,sdm2,cnt2,ftt2,mf3,sdm3,cnt3,ftt3,mf4,sdm4,cnt4,ftt4,mf5,sdm5,cnt5,ftt5*}

$\boldsymbol{I}^{\mathbf{CONS}}$ – technologies having consumables

{*mf1,ftt1,mf2,ftt2,mf3,ftt3,mf4,ftt4,mf5,ftt5,blc*}

$\boldsymbol{I}^{\boldsymbol{BV}}$ – technologies having binary selection constraints

{*htt,slb,ddg,mf1,flc,sdm1,cnt1,ftt1,mf2,sdm2,cnt2,prc,dst,ftt2,mf3,sdm3,cnt3,ftt3,mf4,sdm4,cnt4,ftt4,mf5,sdm5,cnt5,ftt5,blc,dry,byp1,byp2,byp3,byp4*}

$\boldsymbol{I}^{\boldsymbol{S1}}$ – technologies in stage I

{*htt,slb,ddg,mf1,flc,sdm1,cnt1,ftt1,mf3,sdm3,cnt3,ftt3,mf2,sdm2,cnt2,ftt2,byp1,mf4,sdm4,cnt4,ftt4,byp2*}

$\boldsymbol{I}^{\boldsymbol{S2}}$ – technologies in stage II

{*prc,dst,mf5,sdm5,cnt5,ftt5*}

$\boldsymbol{I}^{\boldsymbol{S3}}$ – technologies in stage III

{*blc,byp3,dry,byp4*}

C.3. GAMS optimization model with technology equations:

Annualized cost function derived from (Ulrich and Vasudevan 2004; Biegler, Grossmann, and Westerberg 1997)

```
Positive Variables
         M(j,k)         mass flowrate of component k in stream j
         Xm(jdst,kdst)  mole fractions for distillation columns
         F(jdst,kdst)   molar flowrates in distillation
         CF(i2)         concentration factor of unit i
         Qc(i1)         standard for cost estimation of unit i
         Cc(i1)         purchase cost of unit i
         PW(i1)         power required in unit i
         Mcw(i1)        cooling water required
         Mstm(i1)       steam required for heating
         Nlbr(i1)       number of laborers for unit i
         Cpur(kadd)     purchase cost of added components
         Cons(i4)       consumable cost for membrane units;

*single variables
Positive variables
         QW               Flowrate of aqueous broth (kg-hr-1)
         Feed_C           Cost of entering feed in the system;

Binary variables
         y(ibv)           binary variable corresponding to the unit ibv;

Variable
         Obj              Objective function;

Equations
*Initial flow assignment
InitM1_W initial M1 water flow assignment
InitM1_OthW(kOthW) initial M1 biomass flow assignment
InitFeed initial feed cost
*Component balances in all units
CMB(i,k) component mass balance in all units
*Costs, labor and consumables
Cost_units(i1) cost of units
Nlabr_units(i1) labors required in units
RM_cons_units(i4) consumable costs;

InitM1_W.. M('1','W')=e=Den('W')*QW;
InitM1_OthW(kOthW).. M('1',kOthW)=e=Ttr(kOthW)*QW;
InitFeed.. Feed_C=e=M('1','HLA')*Cfeed;
CMB(i,k)$(kI(i,k)).. sum(j$jIin(j,i),M(j,k))=e=sum(j$jIout(j,i),M(j,k));
Cost_units(i1).. Cc(i1)*(Q0(i1)**nc(i1))=e=C0(i1)*(Qc(i1)**nc(i1));
Nlabr_units(i1).. Nlbr(i1)*Q0(i1)=e=Nlabr(i1)*Qc(i1);
RM_cons_units(i4).. Cons(i4)=e=((Tann/RepT(i4))*CPM(i4)*Qc(i4))*(10**(-6));

Equations
*Logical equations for units to be selected (ibv)
logibv(ibv,j,k) stream flows in binary units
logiddg(j) logical equation for digestion agent in ddg unit
logislb(j) logical equation for solvent in slb unit
logiprc(j) logical equation for anti-solvent in prc unit;

logibv(ibv,j,k)$(jI(j,ibv) and kI(ibv,k) and kJ(j,k)).. M(j,k)-M1(k)*y(ibv)=l=0;
logiddg(j)..  M(j,'DAgnt')-M1('DAgnt')*y('ddg')=l=0;
logislb(j)..  M(j,'Solv')-M1('Solv')*y('slb')=l=0;
logiprc(j)..  M(j,'AnSol')-M1('AnSol')*y('prc')=l=0;

*Stage-I Cell and product isolation technologies:
*Unit selection constraints
Equations
```

```
*unit selection
select1 selection of units in stage-1
selectflcmf selection of flc or mf1
select1flc selection of unit after flc
select2 selection of the 2nd cell removal units
select3 selection of cell removal units after ddg
select4 selection of 2nd set of units after ddg;

select1.. y('htt')+y('slb')+y('ddg')=e=1;
selectflcmf.. y('flc')+y('mf1')=e=y('htt')+y('slb');
select1flc.. y('sdm1')+y('cnt1')+y('ftt1')=e=y('flc');
select2.. y('mf2')+y('sdm2')+y('cnt2')+y('ftt2')+y('byp1')=e=y('htt')+y('slb');
select3.. y('mf3')+y('sdm3')+y('cnt3')+y('ftt3')=e=y('ddg');
select4.. y('mf4')+y('sdm4')+y('cnt4')+y('ftt4')+y('byp2')=e=y('ddg');

y.fx('htt')=0;
*y.fx('cnt3')=1;y.fx('cnt4')=1;
y.fx('ddg')=1;
y.fx('dry')=1;y.fx('blc')=1;

Positive variables Qh_htt;

Equations
*Heat treatment - Utlity costs:
Heat_supp heat supplied to reach the desired temperature
Htt_Stm_req steam required in htt;

Heat_supp.. Qh_htt=e=sum(k1,(M('2',k1)))*Cp('W')*(Tht_out-Tht_in);
Htt_Stm_req.. Mstm('htt')*Hvap_St=e=Qh_htt;

Equations
*Solubilization unit model
Sol_addn amount of solvent added
Cost_sol cost of solvent per hour
SCap_slb_tank Std capacity of solubilization
Pow_slb_tank power required in solubilization tank;

Sol_addn.. M('6','Solv')=e=M('3','HLA')*Sol_add;
Cost_sol.. Cpur('Solv')=e=Cpr('Solv')*M('6','Solv');
SCap_slb_tank.. Qc('slb')=e=(sum(k1,(M('3',k1)/Den(k1))))+(M('6','Solv')/Den('Solv'));
Pow_slb_tank.. PW('slb')=e=Wsp('slb')*Qc('slb');
M.fx('6',kB)=0;M.fx('6','DAgnt')=0;M.fx('6','AnSol')=0;

Equations
*Differential digestion unit model
Agnt_addn amount of agent added
Cost_agnt cost of agent per hour
SCap_ddg_tank standard capacity diff-digestion tank
Pow_ddg_tank power required in diff-digestion tank;

Agnt_addn.. M('8','DAgnt')=e=((sum(kOthW,M('4',kOthW)))-M('4','HLA'))*Agnt_add;
Cost_agnt.. Cpur('DAgnt')=e=Cpr('DAgnt')*M('8','DAgnt');
SCap_ddg_tank.. Qc('ddg')=e=(sum(k1,(M('4',k1)/Den(k1))))+(M('8','DAgnt')/Den('DAgnt'));
Pow_ddg_tank.. PW('ddg')=e=Wsp('ddg')*Qc('ddg');
M.fx('8',kB)=0;M.fx('8','Solv')=0;M.fx('8','AnSol')=0;

Equations
*Microfiltration1 model (mf1)
Retfac_mf1(k1) ret factor for Microfiltration1
Confac_mf1 CF equation for Microfiltration1
Confacmf1_lb CF lb Microfiltration1
Confacmf1_up CF up Microfiltration1
Flux_mf1 flux balance in Microfiltration1
Pow_mf1 power required in Microfiltration1;
```

```
Retfac_mf1(k1).. M('13',k1)=e=M('11',k1)*RCmf1(k1);
Confac_mf1..
CF('mf1')*(sum(k$kJ('14',k),(M('14',k)/Den(k))))=e=(sum(k$kJ('11',k),(M('11',k)/Den(k))));
Confacmf1_lb.. CF('mf1')=g=1.01*y('mf1');
Confacmf1_up.. CF('mf1')=l=35*y('mf1');
Flux_mf1.. J_mf*Qc('mf1')*CF('mf1')=e=(sum(k$kJ('11',k),(M('11',k)/Den(k))))*(CF('mf1')-1);
Pow_mf1.. PW('mf1')=e=Wsp('mf1')*Qc('mf1');

Equations
*Flocculation tank - broth pretreatment/conditioning
Floc_add flocculant mass balance
Floc_cost cost of flocculant added
Vol_floc volume of flocculation tank
Power_floc power required in flocculation;

Floc_add.. M('15','Flcnt')=e=flc_vol*(sum(k$kJ('12',k),(M('12',k)/Den(k))));
Floc_cost.. Cpur('Flcnt')=e=M('15','Flcnt')*CPr('Flcnt');
Vol_floc.. Qc('flc')=e=(sum(k$kJ('12',k),(M('12',k)/Den(k))))*tR_flc;
Power_floc.. PW('flc')=e=Wsp('flc')*(sum(k$kJ('12',k),(M('12',k)/Den(k))));
M.fx('15',k1)=0;

Equations
effeqn_sdm1H efficiency(heavy C) of sdm1 tank
effeqn_sdm1L efficiency(light C) of sdm1 tank
Confac_sdm1 CF equation for sdm1
Confacsdm_lb1 CF sdm1 lb
Confacsdm_up1 CF sdm1 up
*Design equations
Area_sdm1 equation for area sdm1 tank;

effeqn_sdm1H..
eff_sdm*(sum(kHS$kJ('17',kHS),M('17',kHS)))=e=(sum(kHS$kJ('20',kHS),M('20',kHS)))*100;
effeqn_sdm1L..
eff_sdm*(sum(kLS$kJ('17',kLS),M('17',kLS)))=e=(sum(kLS$kJ('21',kLS),M('21',kLS)))*100;
Confac_sdm1..
CF('sdm1')*(sum(k$kJ('21',k),(M('21',k)/Den(k))))=e=(sum(k$kJ('17',k),(M('17',k)/Den(k))));
Confacsdm_lb1.. CF('sdm1')=g=1.1*y('sdm1');
Confacsdm_up1.. CF('sdm1')=l=15*y('sdm1');
*Design equations
Area_sdm1.. (SOR_sdm*3600)*Qc('sdm1')=e=(sum(k$kJ('17',k),(M('17',k)/Den(k))));
PW.fx('sdm1')=0;

Equations
*Centrifugation
Eff_eqn1H efficiency of cnt1 (heavy)
Eff_eqn1L efficiency of cnt1 (light)
Confac_cf1 CF equation for cnt1
Confaccf_lb1 CF centrifuge1 lb
Confaccf_up1 CF centrifuge1 up
*Design equations
Sigmacf1_eqn centrifugation 1 coefficient equation
Power_cf1 power required in centrifuge1,
CW_req_cf1 cooling water required in cnt1 ;

Eff_eqn1H..
eff_cnt*(sum(kHS$kJ('18',kHS),M('18',kHS)))=e=(sum(kHS$kJ('22',kHS),M('22',kHS)))*100;
Eff_eqn1L..
eff_cnt*(sum(kLS$kJ('18',kLS),M('18',kLS)))=e=(sum(kLS$kJ('23',kLS),M('23',kLS)))*100;
Confac_cf1..
CF('cnt1')*(sum(k$kJ('23',k),(M('23',k)/Den(k))))=e=(sum(k$kJ('18',k),(M('18',k)/Den(k))));
Confaccf_lb1.. CF('cnt1')=g=1.1*y('cnt1');
Confaccf_up1.. CF('cnt1')=l=20*y('cnt1');
Sigmacf1_eqn.. Ug_Floc*Qc('cnt1')*3600=e=(sum(k$kJ('18',k),(M('18',k)/Den(k))));
Power_cf1.. PW('cnt1')=e=Wsp('cnt1')*(sum(k$kJ('18',k),(M('18',k)/Den(k))));
CW_req_cf1.. Mcw('cnt1')*Cp('W')*(Tcw_out-Tcw_in)=e=(0.4*PW('cnt1'))*3600;
```

```
Equations
*Filtration
Retfac_ftt1(k1) ret factor for filtration1
Confac_ftt1 CF equation for filtration1
Confacftt1_lb CF filtration lb 1
Confacftt1_up CF filtration up 1
Flux_bal1 flux balance in filtration1
Pow_ftt1 power required for ftt1;

Retfac_ftt1(k1).. M('24',k1)=e=M('19',k1)*RCftt1(k1);
Confac_ftt1..
CF('ftt1')*(sum(k$kJ('25',k),(M('25',k)/Den(k))))=e=(sum(k$kJ('19',k),(M('19',k)/Den(k))));
Confacftt1_lb.. CF('ftt1')=g=1.1*y('ftt1');
Confacftt1_up.. CF('ftt1')=l=30*y('ftt1');
Flux_bal1.. J_ftt*Qc('ftt1')*CF('ftt1')=e=(sum(k$kJ('19',k),(M('19',k)/Den(k))))*(CF('ftt1')-
1);
Pow_ftt1.. PW('ftt1')=e=Wsp('ftt1')*Qc('ftt1');

*StageI - Second cell removal (if required)
Equations
*Microfiltration2 model (mf2)
Retfac_mf2(k1) ret factor for Microfiltration2
Confac_mf2 CF equation for Microfiltration2
Confacmf2_lb CF lb Microfiltration2
Confacmf2_up CF up Microfiltration2
Flux_mf2 flux balance in Microfiltration2
Pow_mf2 power required in Microfiltration2;

Retfac_mf2(k1).. M('32',k1)=e=M('27',k1)*RCmf2(k1);
Confac_mf2..
CF('mf2')*(sum(k$kJ('33',k),(M('33',k)/Den(k))))=e=(sum(k$kJ('27',k),(M('27',k)/Den(k))));
Confacmf2_lb.. CF('mf2')=g=1.01*y('mf2');
Confacmf2_up.. CF('mf2')=l=35*y('mf2');
Flux_mf2.. J_mf*Qc('mf2')*CF('mf2')=e=(sum(k$kJ('27',k),(M('27',k)/Den(k))))*(CF('mf2')-1);
Pow_mf2.. PW('mf2')=e=Wsp('mf2')*Qc('mf2');

Equations
effeqn_sdm2H efficiency(heavy C) of sdm2 tank
effeqn_sdm2L efficiency(light C) of sdm2 tank
Confac_sdm2 CF equation for sdm2
Confacsdm_lb2 CF sdm2 lb
Confacsdm_up2 CF sdm2 up
*Design equations
Area_sdm2 equation for area sdm2 tank;

effeqn_sdm2H..
eff_sdm*(sum(kHS$kJ('28',kHS),M('28',kHS)))=e=(sum(kHS$kJ('34',kHS),M('34',kHS)))*100;
effeqn_sdm2L..
eff_sdm*(sum(kLS$kJ('28',kLS),M('28',kLS)))=e=(sum(kLS$kJ('35',kLS),M('35',kLS)))*100;
Confac_sdm2..
CF('sdm2')*(sum(k$kJ('35',k),(M('35',k)/Den(k))))=e=(sum(k$kJ('28',k),(M('28',k)/Den(k))));
Confacsdm_lb2.. CF('sdm2')=g=1.01*y('sdm2');
Confacsdm_up2.. CF('sdm2')=l=15*y('sdm2');
*Design equations
Area_sdm2.. (SOR_sdm*3600)*Qc('sdm2')=e=(sum(k$kJ('28',k),(M('28',k)/Den(k))));
PW.fx('sdm2')=0;

Equations
*Centrifugation
Eff_eqn2H efficiency of cnt2
Eff_eqn2L efficiency of cnt2
Confac_cf2 CF equation for cnt2
Confaccf_lb2 CF centrifuge2 lb
Confaccf_up2 CF centrifuge2 up
```

```
*Design equations
Sigmacf2_eqn centrifugation 2 coefficient equation
Power_cf2 power required in centrifuge2,
CW_req_cf2 cooling water required in cnt2;

Eff_eqn2H..
eff_cnt*(sum(kHS$kJ('29',kHS),M('29',kHS)))=e=(sum(kHS$kJ('36',kHS),M('36',kHS)))*100;
Eff_eqn2L..
eff_cnt*(sum(kLS$kJ('29',kLS),M('29',kLS)))=e=(sum(kLS$kJ('37',kLS),M('37',kLS)))*100;
Confac_cf2..
CF('cnt2')*(sum(k$kJ('37',k),(M('37',k)/Den(k))))=e=(sum(k$kJ('29',k),(M('29',k)/Den(k))));
Confaccf_lb2.. CF('cnt2')=g=1.01*y('cnt2');
Confaccf_up2.. CF('cnt2')=l=20*y('cnt2');
Sigmacf2_eqn.. Ug_Floc*Qc('cnt2')*3600=e=(sum(k$kJ('29',k),(M('29',k)/Den(k))));
Power_cf2.. PW('cnt2')=e=Wsp('cnt2')*(sum(k$kJ('29',k),(M('29',k)/Den(k))));
CW_req_cf2.. Mcw('cnt2')*Cp('W')*(Tcw_out-Tcw_in)=e=(0.4*PW('cnt2'))*3600;

Equations
*Filtration
Retfac_ftt2(k1) ret factor for filtration2
Confac_ftt2 CF equation for filtration2
Confacftt2_lb CF filtration lb 2
Confacftt2_up CF filtration up 2
Flux_bal2 flux balance in filtration2
Pow_ftt2 power required for ftt2;

Retfac_ftt2(k1).. M('38',k1)=e=M('30',k1)*RCftt2(k1);
Confac_ftt2..
CF('ftt2')*(sum(k$kJ('39',k),(M('39',k)/Den(k))))=e=(sum(k$kJ('30',k),(M('30',k)/Den(k))));
Confacftt2_lb.. CF('ftt2')=g=1.1*y('ftt2');
Confacftt2_up.. CF('ftt2')=l=30*y('ftt2');
Flux_bal2.. J_ftt*Qc('ftt2')*CF('ftt2')=e=(sum(k$kJ('30',k),(M('30',k)/Den(k))))*(CF('ftt2')-
1);
Pow_ftt2.. PW('ftt2')=e=Wsp('ftt2')*Qc('ftt2');

*Stage-I Technology selection after Differential Digestion
Equations
*Microfiltration1 model (mf3)
Retfac_mf3(k1) ret factor for Microfiltration3
Confac_mf3 CF equation for Microfiltration3
Confacmf3_lb CF lb Microfiltration3
Confacmf3_up CF up Microfiltration3
Flux_mf3 flux balance in Microfiltration3
Pow_mf3 power required in Microfiltration3;

Retfac_mf3(k1).. M('47',k1)=e=M('42',k1)*RCmf3(k1);
Confac_mf3..
CF('mf3')*(sum(k$kJ('47',k),(M('47',k)/Den(k))))=e=(sum(k$kJ('42',k),(M('42',k)/Den(k))));
Confacmf3_lb.. CF('mf3')=g=1.01*y('mf3');
Confacmf3_up.. CF('mf3')=l=35*y('mf3');
Flux_mf3.. J_mf*Qc('mf3')*CF('mf3')=e=(sum(k$kJ('42',k),(M('42',k)/Den(k))))*(CF('mf3')-1);
Pow_mf3.. PW('mf3')=e=Wsp('mf3')*Qc('mf3');

Equations
effeqn_sdm3H efficiency(heavy C) of sdm3 tank
effeqn_sdm3L efficiency(light C) of sdm3 tank
Confac_sdm3 CF equation for sdm3
Confacsdm_lb3 CF sdm3 lb
Confacsdm_up3 CF sdm3 up
*Design equations
Area_sdm3 equation for area sdm3 tank;

effeqn_sdm3H..
eff_sdm*(sum(kHD$kJ('43',kHD),M('43',kHD)))=e=(sum(kHD$kJ('49',kHD),M('49',kHD)))*100;
```

```
effeqn_sdm3L..
eff_sdm*(sum(kLD$kJ('43',kLD),M('43',kLD)))=e=(sum(kLD$kJ('48',kLD),M('48',kLD)))*100;
Confac_sdm3..
CF('sdm3')*(sum(k$kJ('49',k),(M('49',k)/Den(k))))=e=(sum(k$kJ('43',k),(M('43',k)/Den(k))));
Confacsdm_lb3.. CF('sdm3')=g=1.1*y('sdm3');
Confacsdm_up3.. CF('sdm3')=l=15*y('sdm3');
*Design equations
Area_sdm3.. (SOR_sdm*3600)*Qc('sdm3')=e=(sum(k$kJ('43',k),(M('43',k)/Den(k))));
PW.fx('sdm3')=0;

Equations
*Centrifugation
Eff_eqn3H efficiency of cnt3 (heavy)
Eff_eqn3L efficiency of cnt3 (light)
Confac_cf3 CF equation for cnt3
Confaccf_lb3 CF centrifuge3 lb
Confaccf_up3 CF centrifuge3 up
*Design equations
Sigmacf3_eqn centrifugation3 coefficient equation
Power_cf3 power required in centrifuge3,
CW_req_cf3 cooling water required in cnt3;

Eff_eqn3H..
eff_cnt*(sum(kHD$kJ('44',kHD),M('44',kHD)))=e=(sum(kHD$kJ('51',kHD),M('51',kHD)))*100;
Eff_eqn3L..
eff_cnt*(sum(kLD$kJ('44',kLD),M('44',kLD)))=e=(sum(kLD$kJ('50',kLD),M('50',kLD)))*100;
Confac_cf3..
CF('cnt3')*(sum(k$kJ('51',k),(M('51',k)/Den(k))))=e=(sum(k$kJ('44',k),(M('44',k)/Den(k))));
Confaccf_lb3.. CF('cnt3')=g=1.1*y('cnt3');
Confaccf_up3.. CF('cnt3')=l=20*y('cnt3');
Sigmacf3_eqn.. Ug_Floc*Qc('cnt3')*3600=e=(sum(k$kJ('44',k),(M('44',k)/Den(k))));
Power_cf3.. PW('cnt3')=e=Wsp('cnt3')*(sum(k$kJ('44',k),(M('44',k)/Den(k))));
CW_req_cf3.. Mcw('cnt3')*Cp('W')*(Tcw_out-Tcw_in)=e=(0.4*PW('cnt3'))*3600;

Equations
*Filtration
Retfac_ftt3(k1) ret factor for filtration3
Confac_ftt3 CF equation for filtration3
Confacftt3_lb CF filtration lb 3
Confacftt3_up CF filtration up 3
Flux_bal3 flux balance in filtration3
Pow_ftt3 power required for ftt3;

Retfac_ftt3(k1).. M('53',k1)=e=M('45',k1)*RCftt3(k1);
Confac_ftt3..
CF('ftt3')*(sum(k$kJ('53',k),(M('53',k)/Den(k))))=e=(sum(k$kJ('45',k),(M('45',k)/Den(k))));
Confacftt3_lb.. CF('ftt3')=g=1.1*y('ftt3');
Confacftt3_up.. CF('ftt3')=l=30*y('ftt3');
Flux_bal3..  J_ftt*Qc('ftt3')*CF('ftt3')=e=(sum(k$kJ('45',k),(M('45',k)/Den(k))))*(CF('ftt3')-
1);
Pow_ftt3.. PW('ftt3')=e=Wsp('ftt3')*Qc('ftt3');

*StageI - Second cell removal (if required)
Equations
*Microfiltration4 model (mf4)
Retfac_mf4(k1) ret factor for Microfiltration4
Confac_mf4 CF equation for Microfiltration4
Confacmf4_lb CF lb Microfiltration4
Confacmf4_up CF up Microfiltration4
Flux_mf4 flux balance in Microfiltration4
Pow_mf4 power required in Microfiltration4;

Retfac_mf4(k1).. M('61',k1)=e=M('55',k1)*RCmf4(k1);
Confac_mf4..
CF('mf4')*(sum(k$kJ('61',k),(M('61',k)/Den(k))))=e=(sum(k$kJ('55',k),(M('55',k)/Den(k))));
```

```
Confacmf4_lb.. CF('mf4')=g=1.01*y('mf4');
Confacmf4_up.. CF('mf4')=l=35*y('mf4');
Flux_mf4.. J_mf*Qc('mf4')*CF('mf4')=e=(sum(k$kJ('55',k),(M('55',k)/Den(k))))*(CF('mf4')-1);
Pow_mf4.. PW('mf4')=e=Wsp('mf4')*Qc('mf4');

Equations
effeqn_sdm4H efficiency(heavy C) of sdm4 tank
effeqn_sdm4L efficiency(light C) of sdm4 tank
Confac_sdm4 CF equation for sdm4
Confacsdm_lb4 CF sdm4 lb
Confacsdm_up4 CF sdm4 up
*Design equations
Area_sdm4 equation for area sdm4 tank;

effeqn_sdm4H..
eff_sdm*(sum(kHD$kJ('56',kHD),M('56',kHD)))=e=(sum(kHD$kJ('63',kHD),M('63',kHD)))*100;
effeqn_sdm4L..
eff_sdm*(sum(kLD$kJ('56',kLD),M('56',kLD)))=e=(sum(kLD$kJ('62',kLD),M('62',kLD)))*100;
Confac_sdm4..
CF('sdm4')*(sum(k$kJ('63',k),(M('63',k)/Den(k))))=e=(sum(k$kJ('56',k),(M('56',k)/Den(k))));
Confacsdm_lb4.. CF('sdm4')=g=1.01*y('sdm4');
Confacsdm_up4.. CF('sdm4')=l=15*y('sdm4');
*Design equations
Area_sdm4.. (SOR_sdm*3600)*Qc('sdm4')=e=(sum(k$kJ('56',k),(M('56',k)/Den(k))));
PW.fx('sdm4')=0;

Equations
*Centrifugation
Eff_eqn4H efficiency of cnt4
Eff_eqn4L efficiency of cnt4
Confac_cf4 CF equation for cnt4
Confaccf_lb4 CF centrifuge4 lb
Confaccf_up4 CF centrifuge4 up
*Design equations
Sigmacf4_eqn centrifugation4 coefficient equation
Power_cf4 power required in centrifuge4,
CW_req_cf4 cooling water required in cnt4;

Eff_eqn4H..
eff_cnt*(sum(kHD$kJ('57',kHD),M('57',kHD)))=e=(sum(kHD$kJ('65',kHD),M('65',kHD)))*100;
Eff_eqn4L..
eff_cnt*(sum(kLD$kJ('57',kLD),M('57',kLD)))=e=(sum(kLD$kJ('64',kLD),M('64',kLD)))*100;
Confac_cf4..
CF('cnt4')*(sum(k$kJ('65',k),(M('65',k)/Den(k))))=e=(sum(k$kJ('57',k),(M('57',k)/Den(k))));
Confaccf_lb4.. CF('cnt4')=g=1.01*y('cnt4');
Confaccf_up4.. CF('cnt4')=l=20*y('cnt4');
Sigmacf4_eqn.. Ug_Floc*Qc('cnt4')*3600=e=(sum(k$kJ('57',k),(M('57',k)/Den(k))));
Power_cf4.. PW('cnt4')=e=Wsp('cnt4')*(sum(k$kJ('57',k),(M('57',k)/Den(k))));
CW_req_cf4.. Mcw('cnt4')*Cp('W')*(Tcw_out-Tcw_in)=e=(0.4*PW('cnt4'))*3600;

Equations
*Filtration
Retfac_ftt4(k1) ret factor for filtration4
Confac_ftt4 CF equation for filtration4
Confacftt4_lb CF filtration lb 4
Confacftt4_up CF filtration up 4
Flux_bal4 flux balance in filtration4
Pow_ftt4 power required for ftt4;

Retfac_ftt4(k1).. M('67',k1)=e=M('58',k1)*RCftt4(k1);
Confac_ftt4..
CF('ftt4')*(sum(k$kJ('67',k),(M('67',k)/Den(k))))=e=(sum(k$kJ('58',k),(M('58',k)/Den(k))));
Confacftt4_lb.. CF('ftt4')=g=1.1*y('ftt4');
Confacftt4_up.. CF('ftt4')=l=30*y('ftt4');
```

```
Flux_bal4.. J_ftt*Qc('ftt4')*CF('ftt4')=e=(sum(k$kJ('58',k),(M('58',k)/Den(k))))*(CF('ftt4')-
1);
Pow_ftt4.. PW('ftt4')=e=Wsp('ftt4')*Qc('ftt4');

*M.up('70','HLA')=20;

*Stage-II Product Concentration
Equations
Selectstg2 selection of prc unit
Selectdst selection of dst unit
Selectsep2 selection of separation units after prc;

Selectstg2.. y('prc')=e=y('htt')+y('slb');
Selectdst.. y('dst')=e=y('prc');
Selectsep2.. y('mf5')+y('sdm5')+y('cnt5')+y('ftt5')=e=y('prc');

*$ontext
*Precipitation model
Equations
*Prc_eff(k1) precipitation efficiency
Ansl_addn amount of Ansl added
Cost_Ansl cost of Ansl per hour
SCap_ppt_tank cost of pptn tank
Pow_ppt_tank power required in pptn tank;

*Prc_eff(k1).. M('71',k1)=e=Eprc(k1)*M('41',k1);
Ansl_addn.. M('70','AnSol')=e=M('41','HLA')*AnSl_add;
Cost_Ansl.. Cpur('AnSol')=e=Cpr('AnSol')*M('87','AnSol');
SCap_ppt_tank.. Qc('prc')=e=(sum(k1,(M('41',k1)/Den(k1))))+sum(k1,(M('70',k1)/Den(k1)));
Pow_ppt_tank.. PW('prc')=e=Wsp('prc')*Qc('prc');
M.fx('87',kB)=0;M.fx('87','Solv')=0;M.fx('87','DAgnt')=0;

*Stage-II Technology selection after Precipitation
Equations
*Microfiltration5 model (mf5)
Retfac_mf5(k1) ret factor for Microfiltration5
Confac_mf5 CF equation for Microfiltration5
Confacmf5_lb CF lb Microfiltration5
Confacmf5_up CF up Microfiltration5
Flux_mf5 flux balance in Microfiltration5
Pow_mf5 power required in Microfiltration5;

Retfac_mf5(k1).. M('77',k1)=e=M('72',k1)*RCmf5(k1);
Confac_mf5..
CF('mf5')*(sum(k$kJ('77',k),(M('77',k)/Den(k))))=e=(sum(k$kJ('72',k),(M('72',k)/Den(k))));
Confacmf5_lb.. CF('mf5')=g=1.0001*y('mf5');
Confacmf5_up.. CF('mf5')=l=35*y('mf5');
Flux_mf5.. J_mf*Qc('mf5')*CF('mf5')=e=(sum(k$kJ('72',k),(M('72',k)/Den(k))))*(CF('mf5')-1);
Pow_mf5.. PW('mf5')=e=Wsp('mf5')*Qc('mf5');

Equations
effeqn_sdm5H efficiency(heavy C) of sdm5 tank
effeqn_sdm5L efficiency(light C) of sdm5 tank
Confac_sdm5 CF equation for sdm5
Confacsdm_lb5 CF sdm5 lb
Confacsdm_up5 CF sdm5 up
*Design equations
Area_sdm5 equation for area sdm5 tank;

effeqn_sdm5H..
eff_sdm*(sum(kHP$kJ('73',kHP),M('73',kHP)))=e=(sum(kHP$kJ('79',kHP),M('79',kHP)))*100;
effeqn_sdm5L..
eff_sdm*(sum(kLP$kJ('73',kLP),M('73',kLP)))=e=(sum(kLP$kJ('78',kLP),M('78',kLP)))*100;
Confac_sdm5..
CF('sdm5')*(sum(k$kJ('79',k),(M('79',k)/Den(k))))=e=(sum(k$kJ('73',k),(M('73',k)/Den(k))));
```

```
Confacsdm_lb5.. CF('sdm5')=g=1.0001*y('sdm5');
Confacsdm_up5.. CF('sdm5')=l=15*y('sdm5');
*Design equations
Area_sdm5.. (SOR_sdm*3600)*Qc('sdm5')=e=(sum(k$kJ('73',k),(M('73',k)/Den(k))));
PW.fx('sdm5')=0;

Equations
*Centrifugation
Eff_eqn5H efficiency of cnt5 (heavy)
Eff_eqn5L efficiency of cnt5 (light)
Confac_cf5 CF equation for cnt5
Confaccf_lb5 CF centrifuge5 lb
Confaccf_up5 CF centrifuge5 up
*Design equations
Sigmacf5_eqn centrifugation5 coefficient equation
Power_cf5 power required in centrifuge5,
CW_req_cf5 cooling water required in cnt5;

Eff_eqn5H..
eff_cnt*(sum(kHP$kJ('74',kHP),M('74',kHP)))=e=(sum(kHP$kJ('81',kHP),M('81',kHP)))*100;
Eff_eqn5L..
eff_cnt*(sum(kLP$kJ('74',kLP),M('74',kLP)))=e=(sum(kLP$kJ('80',kLP),M('80',kLP)))*100;
Confac_cf5..
CF('cnt5')*(sum(k$kJ('81',k),(M('81',k)/Den(k))))=e=(sum(k$kJ('74',k),(M('74',k)/Den(k))));
Confaccf_lb5.. CF('cnt5')=g=1.0001*y('cnt5');
Confaccf_up5.. CF('cnt5')=l=20*y('cnt5');
Sigmacf5_eqn.. Ug_Floc*Qc('cnt5')*3600=e=(sum(k$kJ('74',k),(M('74',k)/Den(k))));
Power_cf5.. PW('cnt5')=e=Wsp('cnt5')*(sum(k$kJ('74',k),(M('74',k)/Den(k))));
CW_req_cf5.. Mcw('cnt5')*Cp('W')*(Tcw_out-Tcw_in)=e=(0.4*PW('cnt5'))*3600;

Equations
*Filtration5
Retfac_ftt5(k1) ret factor for filtration5
Confac_ftt5 CF equation for filtration5
Confacftt5_lb CF filtration lb5
Confacftt5_up CF filtration up5
Flux_bal5 flux balance in filtration5
Pow_ftt5 power required for ftt5;

Retfac_ftt5(k1).. M('83',k1)=e=M('75',k1)*RCftt5(k1);
Confac_ftt5..
CF('ftt5')*(sum(k$kJ('83',k),(M('83',k)/Den(k))))=e=(sum(k$kJ('75',k),(M('75',k)/Den(k))));
Confacftt5_lb.. CF('ftt5')=g=1.0001*y('ftt5');
Confacftt5_up.. CF('ftt5')=l=30*y('ftt5');
Flux_bal5..  J_ftt*Qc('ftt5')*CF('ftt5')=e=(sum(k$kJ('75',k),(M('75',k)/Den(k))))*(CF('ftt5')-
1);
Pow_ftt5.. PW('ftt5')=e=Wsp('ftt5')*Qc('ftt5');
*$offtext

$ontext
*Model for distillation with unit conversion to moles
Equations
Mol_dst1(jdst,kdst) molar flow of components in dst1
CMB_dst1(kdst) component balance in dst1
Molfrac_dst1(jdst,kdst) molar flows of components;

Mol_dst1(jdst,kdst).. F(jdst,kdst)*MW(kdst)=e=M(jdst,kdst);
CMB_dst1(kdst)..
sum(jdst$jIin('dst',jdst),F(jdst,kdst))=e=sum(jdst$jIout('dst',jdst),F(jdst,kdst));
Molfrac_dst1(jdst,kdst).. Xm(jdst,kdst)*(sum(kkdst,(F(jdst,kkdst))))=e=F(jdst,kdst);

Equations
*Distillation component constraints:
*heavy key component - Water
*light key component - AnSol
```

```
HVthanHK(kdst) heavier than HK not in distillate
LTthanLK(kdst) lighter than LK not in bottoms
RecWater_constr Water recovery constraint1
RecLcoprd_constr Slb component recovery
RecAnSol_constr AnSol recovery constraint1;

HVthanHK(kdst).. Xm('85',kdst)$(RV(kdst) lt RV('W'))=e=0;
LTthanLK(kdst).. Xm('86',kdst)$(RV(kdst) gt RV('AnSol'))=e=0;
RecWater_constr.. Xm('85','W')=l=0.07;
RecLcoprd_constr.. Xm('85','Lcoprd')=l=0.01;
RecAnSol_constr.. Xm('85','AnSol')=e=0.9;

Positive variables
*Design variables
         Qh_sat - heating to reach saturation temperature
         Nmin - minimum number of plates
         N - number of plates
         Nact - actual number of plates
         Rmin - minimum reflux ratio
         Ract - reflux ratio actual
         Uv - Underwoods variable
         L_dt - liquid flowrate in column
         V_dt - vapor flowrate in column
         DC_dt - diameter
         HC_dt - height of column;

Equations
*Design equations
Min_N minimum number of plates-1
UW_eq underwood equation
Min_R minimum reflux ratio eqn
Ref_R reflux ratio
Liq_flow_dt liquid flowrate in distillation column
Vap_flow_dt vapor flowrate in dt
Theo_N theoretical plates
Act_N actual plates
Ht_col height of column
Dt_col diameter of column
Vol_dst volume of column
Dia_Ht_dst Length to diameter ratio dst1;

Min_N..    Nmin*log(RV('AnSol'))=e=log(Xm('85','AnSol'))+log(Xm('86','W'))-log(Xm('85','W'))-
log(Xm('86','AnSol'));
UW_eq.. sum(kdst$(RV(kdst) ne 0),(Xm('84',kdst)*RV(kdst))*(prod(kkdst$(not sameas(kdst,kkdst)
and (RV(kdst) ne 0)),(RV(kkdst)-Uv))))=e=0;
Min_R..   (1+Rmin)*(prod(kdst$(RV(kdst)   ne   0),(RV(kdst)-Uv)))=e=sum(kdst$(RV(kdst)   ne
0),(Xm('85',kdst)*RV(kdst))*(prod(kkdst$(not    sameas(kdst,kkdst)    and    (RV(kdst)    ne
0)),(RV(kkdst)-Uv))));
Ref_R.. Ract=e=Ref_ct*Rmin;
Liq_flow_dt.. L_dt=e=Ract*(sum(kdst,(M('85',kdst))));
Vap_flow_dt.. V_dt=e=L_dt+(sum(kdst,(M('85',kdst))));
*Theo_N1..                    (N1-N1min)*((R1act+1)**0.57)=e=(0.75*(N1+1)*((R1act+1)**0.57))-
(0.75*(N1+1)*((R1act-R1min)**0.57));
*To use the previous Gilliland equation we need either N1min or R1min to be specified
Theo_N.. N*0.6=e=Nmin;
*This equation is taken from Sinnot and Towler (RC-6)
Act_N.. Nact*Eff_stg=e=N;
Ht_col.. HC_dt=e=H_stg*Nact;
Dt_col.. (DC_dt**2)*pi*u_vap*3600*Den('AnSol')=g=4*(V_dt);
Vol_dst.. 4*Qc('dst')=e=(pi*(DC_dt**2)*HC_dt);
Dia_Ht_dst.. HC_dt=l=DC_dt*30;

*Variable bounds
Equations
Nstage_lb number of stages LB
```

```
RefR_lb reflux ratio LB;
Nstage_lb.. Nmin=g=1*y('dst');
RefR_lb.. Rmin=g=1.01*y('dst');

Xm.l(jdst,kdst)=0.0001;
Ract.l=1.1;
Uv.l=2.875;
N.l=5;

Positive variables
Qcdt - condensor duty
Qhdt - reboiler duty
Mcwt - Mass of cooling water
Mstm - Mass of steam;

*Utlity costs:
Equations
Heat_sat heating to saturation temperature
Ht_dist heat duty in dst-1
Cl_dist cooling in dst-1
Stm_req steam required in dst-1
Cwt_req cooling water required in dst-1;

Heat_sat.. Qh_sat=e=sum(kdst,(M('84',kdst)))*Cp('AnSol')*(Tsat-Tamb);
Ht_dist.. Qhdt=e=(sum(kdst,(F('85',kdst)*MW(kdst)*Hvap(kdst))))*(Ract+1);
Cl_dist.. Qcdt=e=(sum(kdst,(F('85',kdst)*MW(kdst)*Hvap(kdst))))*Ract;
Stm_req.. Mstm('dst')*Hvap_St=e=Qhdt+Qh_sat;
Cwt_req.. Mcw('dst')*Cp('W')*(Tcw_out-Tcw_in)=e=Qcdt;
$offtext

*Stage-III Product refining
*Logical equations for the unit selection in stage-III:
Equations
*unit selection
select3blc selection of blc unit
select3dry selection of dry unit;
select3blc.. y('blc')+y('byp3')=e=1;
select3dry.. y('dry')+y('byp4')=e=1;

Equations
Bleach_eff efficiency of blc
Prod_blc product in bleaching unit
Blc_volume volume of blc column
Pow_blc power reqd in blc;

Bleach_eff..
sum(kOthP$kJ('92',kOthP),M('92',kOthP))=e=Blc_eff*sum(kOthP$kJ('90',kOthP),M('90',kOthP));
Prod_blc.. M('93','HLA')=e=M('90','HLA');
Blc_volume.. Qc('blc')=e=(sum(k$kJ('90',k),(M('90',k)/Den(k))))*tR_blc;
Pow_blc.. PW('blc')=e=Wsp('blc')*Qc('blc');

*Dryer model (freeze dryer)
Positive variables
Qtd - energy required for sublimation
Mrfg - mass flowrate of refrigerant
A_dry - Area of dryer;

Equations
Sub_liq liquid sublimed in exit stream
HLA_bal cyanophycin in product stream 3
Energy_sub energy required for sublimation
Refg_flow flowrate of refrigerant
Area_dry area of dryer
Sub_Capacity sublimation capacity eqn
Power_dryer power required in a dryer;
```

```
Sub_liq..
sum(kOthP$kJ('98',kOthP),M('98',kOthP))=e=Sub_sol*sum(kOthP$kJ('96',kOthP),M('96',kOthP));
HLA_bal.. M('99','HLA')=e=M('96','HLA');
Energy_sub.. Qtd=e=sum(k1,M('98',k1))*((Cp('Solv')*abs(T_frz-T_amb))+Hsub_sol);
Refg_flow.. Qtd=e=Mrfg*Cp_rfg*abs(Tout_rfg-Tin_rfg);
Area_dry.. Qtd=e=Utdry*A_dry*abs(Tout_rfg-T_amb);
Sub_Capacity.. Qc('dry')=e=sum(k1,M('98',k1));
Power_dryer.. PW('dry')=e=Wsp('dry')*A_dry;

Equations
*product conditions
Prod_pure product purity
Prod_amt amount of final product;

Prod_pure..  M('101','HLA')=g=Purity*sum(k$kJ('101',k),M('101',k));
Prod_amt.. sum(k$kJ('101',k),M('101',k))=e=Prd_F;

*Equations for stagewise cost analysis:
Positive variables
CCAC(Nstg) Annualized capital cost in stages
CCRM(Nstg) Raw material costs in stages
CCCS(Nstg) Consumable cost in stages
CCLB(Nstg) Labor costs in stages
CCUT(Nstg) Utility costs in stages
CCTC(Nstg) Total cost in stages
CCOT(Nstg) Other costs in stages
CFAC annual feed costs;

Equation
ACC_eq_s1,ACC_eq_s2,ACC_eq_s3 Annualized capital cost eqns
RMC_eq_s1,RMC_eq_s2,RMC_eq_s3 Material costs in stages eqns
CSC_eq_s1,CSC_eq_s2,CSC_eq_s3 Consumables costs in stages eqns
LBC_eq_s1,LBC_eq_s2,LBC_eq_s3 Labor costs in stages eqns
UTC_eq_s1,UTC_eq_s2,UTC_eq_s3 Utility costs in stages eqns
TC_eq(Nstg) Total cost in stages eqns
OTC_eq(Nstg) Other costs in stages eqns
FeedC_eq Annual cost of entering feed;

ACC_eq_s1.. CCAC('s1')=e=1.66*CRF*BMC_mult*(sum(istg1,Cc(istg1)))/(Prd_F*Tann);
ACC_eq_s2.. CCAC('s2')=e=1.66*CRF*BMC_mult*(sum(istg2,Cc(istg2)))/(Prd_F*Tann);
ACC_eq_s3.. CCAC('s3')=e=1.66*CRF*BMC_mult*(sum(istg3,Cc(istg3)))/(Prd_F*Tann);

RMC_eq_s1.. CCRM('s1')=e=(Cpur('Flcnt')+Cpur('Solv')+Cpur('DAgnt'))/(Prd_F*(10**6));
RMC_eq_s2.. CCRM('s2')=e=(Cpur('AnSol'))/(Prd_F*(10**6));
RMC_eq_s3.. CCRM('s3')=e=0;

CSC_eq_s1..
CCCS('s1')=e=(Cons('ftt1')+Cons('mf1')+Cons('ftt2')+Cons('mf2')+Cons('ftt3')+Cons('mf3')+Cons
('ftt4')+Cons('mf4'))/(Prd_F*Tann);
CSC_eq_s2.. CCCS('s2')=e=(Cons('ftt5')+Cons('mf5'))/(Prd_F*Tann);
CSC_eq_s3.. CCCS('s3')=e= Cons('blc')/(Prd_F*Tann);

LBC_eq_s1.. CCLB('s1')=e=Clbhr*sum(istg1,Nlbr(istg1))/(Prd_F*(10**6));
LBC_eq_s2.. CCLB('s2')=e=Clbhr*sum(istg2,Nlbr(istg2))/(Prd_F*(10**6));
LBC_eq_s3.. CCLB('s3')=e=Clbhr*sum(istg3,Nlbr(istg3))/(Prd_F*(10**6));

UTC_eq_s1..
CCUT('s1')=e=((C_elec*sum(istg1,PW(istg1)))+(C_cwt*sum(istg1,Mcw(istg1)))+(C_stm*sum(istg1,Ms
tm(istg1)))))/(Prd_F*(10**6));
UTC_eq_s2..
CCUT('s2')=e=((C_elec*sum(istg2,PW(istg2)))+(C_cwt*sum(istg2,Mcw(istg2)))+(C_stm*sum(istg2,Ms
tm(istg2)))))/(Prd_F*(10**6));
```

```
UTC_eq_s3..
CCUT('s3')=e=((C_elec*sum(istg3,PW(istg3)))+(C_cwt*sum(istg3,Mcw(istg3)))+(C_stm*sum(istg3,Ms
tm(istg3)))+(C_rfrg*Mrfg))/(Prd_F*(10**6));

TC_eq(Nstg).. CCTC(Nstg)=e=CCAC(Nstg)+CCRM(Nstg)+CCCS(Nstg)+CCUT(Nstg)+2.78*CCLB(Nstg);
OTC_eq(Nstg)..                                                    CCOT(Nstg)=e=CCTC(Nstg)-
(CCAC(Nstg)+CCRM(Nstg)+CCCS(Nstg)+CCLB(Nstg)+CCUT(Nstg));
FeedC_eq.. CFAC=e=Feed_C/(Prd_F*(10**6));

Positive variables
CCTAC - Annualized capital cost
CCTRM - raw material cost
CCTCS - Consumables cost
CCTUT - utility cost
CCTLB - labor cost
CCTOT - Other costs;

Variable
CCTPC - total process cost;

Equation
ACap_Cost annualized total capital cost
RM_Cost raw materials cost
CS_Cost Consumable cost eqn
Labor_Cost labor cost
Util_Cost utility cost
TPC_fun total process cost
OTH_cost total other costs
Ob_fun objective function;

ACap_Cost.. CCTAC=e=sum(Nstg,CCAC(Nstg));
RM_Cost.. CCTRM=e=sum(Nstg,CCRM(Nstg));
CS_Cost.. CCTCS=e=sum(Nstg,CCCS(Nstg));
Labor_Cost.. CCTLB=e=sum(Nstg,CCLB(Nstg));
Util_Cost.. CCTUT=e=sum(Nstg,CCUT(Nstg));
OTH_cost.. CCTOT=e=sum(Nstg,CCOT(Nstg));
TPC_fun.. CCTPC=e=CCTAC+CCTRM+CCTCS+CCTLB+CCTUT+CCTOT+CFAC;

Ob_fun.. Obj=e=CCTPC;

Model CS_EX_NSL_SLD /all/;
Solve %ModelName% minimizing Obj using %type%;
```

References:

1. Biegler, Lorenz T., Ignacio E. Grossmann, and Arthur W. Westerberg. 1997. *Systematic Methods of Chemical Process Design*. Prentice Hall PTR.
2. Ulrich, Gael D., and Palligarnai T. Vasudevan. 2004. Chemical Engineering Process Design and Economics: A Practical Guide. Process Pub.
3. Yenkie KM, Wu W, Clark RL, Pfleger BF, Root TW, Maravelias CT. A roadmap for the synthesis of separation networks for the recovery of bio-based chemicals: Matching biological and process feasibility. Biotechnol Adv. 2016. doi:10.1016/j.biotechadv.2016.10.003.
4. Kokossis AC, Tsakalova M, Pyrgakis K. Design of integrated biorefineries. Comput Chem Eng. 2015;81:40–56.
5. Wu W, Henao CA, Maravelias CT. A superstructure representation, generation, and modeling framework for chemical process synthesis. AIChE J. 2016. http://onlinelibrary.wiley.com/doi/10.1002/aic.15300/abstract. Accessed 3 Aug 2016.
6. Yeomans H, Grossmann IE. A systematic modeling framework of superstructure optimization in process synthesis. Comput Chem Eng. 1999;23:709–31.
7. Kim J, Sen SM, Maravelias CT. An optimization-based assessment framework for biomass-to-fuel conversion strategies. Energy Environ Sci. 2013;6:1093.
8. Yenkie, Kirti M., Wenzhao Wu, and Christos T. Maravelias. 2017. "Synthesis and Analysis of Separation Networks for the Recovery of Intracellular Chemicals Generated from Microbial-Based Conversions." Biotechnology for Biofuels 10 (May): 119. https://doi.org/10.1186/s13068-017-0804-2.
9. Kılınç, Mustafa R., and Nikolaos V. Sahinidis. 2018. "Exploiting Integrality in the Global Optimization of Mixed-Integer Nonlinear Programming Problems with BARON." Optimization Methods and Software 33 (3): 540–62. https://doi.org/10.1080/10556788.2017.1350178.