Week 3: Modeling II, Networks, Flows and Process Superstructures

2105623 Optimization of Chemical Processes

Assoc. Prof. Dr. Soorathep Kheawhom

Department of Chemical Engineering, Chulalongkorn University

Semester 1/2026

Outline

1. Lecture 1: networks and the node balance
2. Lecture 2: paths, cuts and integrality
3. Lecture 3: process superstructures

Session plan, 180 minutes

Time	Block	Duration
0:00 to 0:10	Opening: recap of Week 2, one question from HW1, today's objectives	10 min
0:10 to 0:40	**Lecture 1**: networks as a modeling language, the node balance, transportation and transshipment	30 min
0:40 to 1:00	**Activity 1**: draw the network (pairs, paper)	20 min
1:00 to 1:30	**Lecture 2**: shortest path, maximum flow, the minimum cut, total unimodularity	30 min
1:30 to 1:45	Break	15 min
1:45 to 2:10	**Activity 2**: transportation in OpenSolver (computer, individual)	25 min
2:10 to 2:35	**Lecture 3**: process superstructures, continuous route splits, where binaries would enter	25 min
2:35 to 2:55	**Activity 3**: sketch a superstructure (groups of 3 to 4)	20 min
2:55 to 3:00	Wrap-up, takeaways, what is due next week	5 min

Tools today: OpenSolver for the matrix models, Pyomo for the graph models. Notebook `W03_modeling2_networks.ipynb`; workbook `W03_transportation_STUDENT.xlsx`.

Learning objectives

After this session you should be able to

1. Write the node balance that every network flow model shares, and recognize the node-arc incidence matrix E behind it.
2. Formulate transportation, transshipment, shortest path and maximum flow as four instances of one minimum cost flow template.
3. Explain why a network LP returns integral flows with no integer variable, using total unimodularity, and state exactly what destroys that property.
4. Read node potentials and arc reduced costs from the LP duals, and identify a minimum cut from a maximum flow solution.
5. Represent competing process routes as a superstructure, and say which decisions are continuous and which would require a binary variable.

CLO 1 (formulation) and CLO 4 (implementation and solution in Pyomo). Reference: Williams, Ch. 5; Rao, Ch. 3; Edgar, Himmelblau and Lasdon, Ch. 7.

From Week 2 patterns to a network

The four reusable patterns of Week 2

1. index set
2. balance equation
3. capacity constraint
4. linking constraint

This week

A network model is those four patterns applied to a graph. The index set is the arc set, the balance equation is written once per node, the capacity constraint is one bound per arc, and the graph itself is the linking structure.

Why networks deserve their own week

- A very large class of engineering problems is a network in disguise: distribution, logistics, pipeline routing, assignment, project scheduling, flowsheet synthesis.
- The structure buys three things: integral answers for free, duals with an immediate physical reading, and specialized algorithms orders of magnitude faster than general LP.
- The node balance is literally a steady-state material balance. Chemical engineers already know this equation.

One template: the minimum cost flow problem

Let $G = (N, A)$ be a directed graph. For arc $(i, j) \in A$ let $f_{ij} \geq 0$ be the flow, c_{ij} the unit cost and u_{ij} the capacity. For node $n \in N$ let b_n be the net supply.

Minimum cost flow

$$\min_{f} \sum_{(i,j)\in A} c_{ij} f_{ij} \quad \text{s.t.} \quad \underbrace{\sum_{j:(n,j)\in A} f_{nj} - \sum_{i:(i,n)\in A} f_{in} = b_n}_{\text{node balance, one per node}} \quad \forall n \in N, \qquad 0 \leq f_{ij} \leq u_{ij}.$$

- $b_n > 0$ at a source, $b_n < 0$ at a sink, $b_n = 0$ at a pure transit node.
- Feasibility of the whole class requires $\sum_n b_n = 0$.
- In matrix form the balance is $Ef = b$, with E the node-arc incidence matrix.

The engineering reading

outflow − inflow = net generation. This is the steady-state material balance around a unit, written once per node instead of once per unit.

The node-arc incidence matrix

Definition

$$E_{n,(i,j)} = \begin{cases} +1 & \text{if } n = i \quad \text{(arc leaves } n\text{)} \\ -1 & \text{if } n = j \quad \text{(arc enters } n\text{)} \\ 0 & \text{otherwise} \end{cases}$$

Every column has exactly one $+1$ and one -1, and therefore sums to zero.

- Rows are nodes, columns are arcs. The whole model is $\min c^{\mathsf{T}} f$ subject to $Ef = b$, $0 \leq f \leq u$.
- The rows are linearly dependent: they sum to the zero vector, which is exactly why $\sum_n b_n = 0$ is required.

Four classical problems, one template

Problem	Graph	Data choice
Transportation	bipartite, sources to sinks	unit shipping cost, no capacities, b from supply and demand
Transshipment	any graph, transit nodes allowed	unit cost, arc limits, $b_n = 0$ at transit nodes
Shortest path	any graph	arc length as cost, $b = +1$ at the origin and -1 at the destination
Maximum flow	any graph, plus a super source and super sink	zero cost except -1 on a return arc, arc limits, $b = 0$

Transportation: the bipartite special case

Model

$$\min \sum_{i \in S} \sum_{j \in D} c_{ij} x_{ij}$$

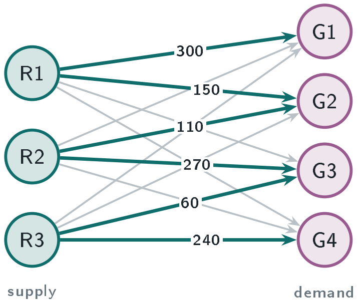
$$\text{s.t.} \quad \sum_{j \in D} x_{ij} \le s_i \qquad \forall i \in S$$

$$\sum_{i \in S} x_{ij} \ge d_j \qquad \forall j \in D$$

$$x_{ij} \ge 0.$$

Three precursor plants $S = \{R1, R2, R3\}$ supply four cell gigafactories $D = \{G1, \ldots, G4\}$. Supplies 450, 380, 300 t/week; demands 300, 260, 330, 240 t/week. Total supply 1,130 t equals total demand, so the instance is balanced.

c_{ij}, USD/t	G1	G2	G3	G4
R1	18	24	31	40
R2	27	19	22	33
R3	38	30	25	21



Teal arcs carry flow (tonnes per week); mist arcs are empty at the optimum.

Transportation: the optimum and what the duals say

Optimum

Minimum shipping cost = 23,570.00 USD/week.

x_{ij}, t	G1	G2	G3	G4
R1	300	150	0	0
R2	0	110	270	0
R3	0	0	60	240

All seven constraints bind: the instance is balanced, so nothing is left over anywhere.

Duals: the classical u-v (MODI) potentials

$$u = (0, -5, -2)\ \text{USD/t}, \qquad v = (18, 24, 27, 23)\ \text{USD/t}$$

$$c_{ij} - u_i - v_j = 0 \ \text{ on every lane carrying flow,}$$

$$c_{ij} - u_i - v_j \geq 0 \ \text{ on every empty lane.}$$

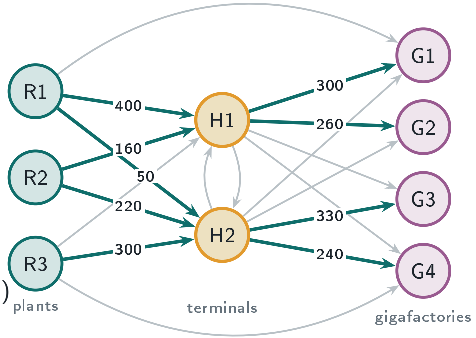
- v_j is the delivered value of one extra tonne at gigafactory j; u_i the value of one extra tonne of capacity at plant i.
- $R1$ has $u_1 = 0$ although its supply row binds: a degenerate vertex, and $R1$ is the marginal plant. This is the Week 8 hook.
- The MODI test comes free from the LP dual. No hand-built tableau is needed.

Transshipment: transit nodes and arc capacities

The same company operates two coastal terminals $H1$ and $H2$. Material moves plant to terminal, terminal to terminal, and terminal to gigafactory. Two direct lanes remain. **Every arc now has a capacity**, and the terminals are pure transit nodes with $b_n = 0$.

9 nodes, 18 arcs.

$b = (450, 380, 300, 0, 0, -300, -260, -330, -240)$



Model: the general template written verbatim, one balance per node and one bound per arc.

Optimal cost = 25,850.00 USD/week.

Optimal flows in t/week. $R1 \to H1$ saturates at 400 t and $R3 \to H2$ at 300 t; both inter-terminal arcs and both direct lanes are empty.

The two totals are not comparable, and that matters

A comparison students always try to make

Transportation 23,570.00 against transshipment 25,850.00 USD/week. The obvious conclusion, that terminals cost money, is **wrong**, because the two models are not two options for the same system.

- The direct problem has *unlimited* lane capacity and no terminal handling.
- The transshipment network puts a capacity on *every* arc and forces most flow through a terminal.
- The comparison that means something is between transshipment options: for example, what widening $R1 \to H1$ is worth.

Node potentials π_n and arc reduced costs

$$\bar{c}_{ij} = c_{ij} - (\pi_i - \pi_j)$$

$\bar{c}_{ij} > 0$	the arc must be empty
$\bar{c}_{ij} < 0$	the arc must be saturated
$\bar{c}_{ij} = 0$	any flow between the bounds

These are the optimality conditions of the minimum cost flow problem, and they are read directly off the LP duals of the node balances.

Modeling rule

State what a model *does not* include before comparing its objective with anything.

Activity 1 | 20 min | pairs, paper: draw the network

The system, in words

A battery-materials company operates **two refineries**, **two consolidation hubs** and supplies **four gigafactories**. Refinery **R1** can ship 450 t/week and refinery **R2** can ship 380 t/week. The gigafactories require **G1** 200, **G2** 190, **G3** 260 and **G4** 180 t/week, so total supply equals total demand at 830 t/week. Refineries ship only to hubs, except for one surviving direct lane from R1 to G1. The two hubs may re-ship to each other in either direction. Each hub may ship to any gigafactory. Hubs hold no stock: whatever enters in a week leaves in the same week.

Arc data: cost in USD/t, capacity in t/week

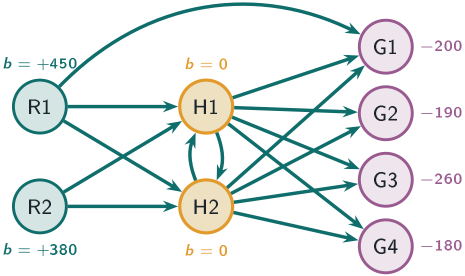
arc	c	u	arc	c	u	arc	c	u
$R1 \to H1$	12	400	$H1 \to H2$	5	200	$H2 \to G1$	22	250
$R1 \to H2$	20	250	$H2 \to H1$	5	200	$H2 \to G2$	13	300
$R2 \to H1$	16	300	$H1 \to G1$	8	350	$H2 \to G3$	9	350
$R2 \to H2$	14	350	$H1 \to G2$	11	300	$H2 \to G4$	12	300
$R1 \to G1$	30	150	$H1 \to G3$	17	250	$H1 \to G4$	26	200

Your task

1. Draw the node-arc diagram by hand. Label every arc with its cost and capacity.
2. Mark each node with its net supply b_n.
3. Write the node balance for **hub H1** in full algebra, in the sign convention outflow minus inflow equals b_n.

Grouping pairs. **Time** 20 min. At 15 min three diagrams are held up and compared. **Hand back** one sheet per pair: the diagram plus the one balance equation.

Activity 1 | [ANSWER] | the balances are the model



8 nodes, 15 arcs, $\sum_n b_n = 830 - 830 = 0$.

Node balance at H1

$$\underbrace{f_{H1,H2} + f_{H1,G1} + f_{H1,G2} + f_{H1,G3} + f_{H1,G4}}_{\text{outflow}} - \underbrace{\left(f_{R1,H1} + f_{R2,H1} + f_{H2,H1}\right)}_{\text{inflow}} = 0.$$

The teaching point

Different pairs drew the same network with different layouts, orderings and crossings. **None of that is the model. The model is the incidence structure: which arcs exist, and which nodes each one joins.** A diagram is a communication device; the balance equations are the mathematics.

Two errors to look for: writing b_{H1} as a supply because the hub is drawn in the middle, and forgetting the reverse arc $H2 \to H1$, which is a *separate* arc with its own non-negative flow, not a negative flow on $H1 \to H2$.

Shortest path is a minimum cost flow

The only change is b

Send exactly one unit from origin to destination, interpret c_{ij} as length, and drop the capacities:

$$b_{\text{origin}} = +1, \quad b_{\text{dest}} = -1, \quad b_n = 0 \;\text{ elsewhere.}$$

Nothing else changes. The optimal flow is a single path, because the incidence matrix forces an integral basic optimal solution and the only integral unit flows are paths.

First payoff of total unimodularity

An inherently combinatorial object, a path, comes out of a linear program.

Worked instance: fastest route $R1 \to G4$

Arc costs re-read as transit times in hours.

Route	hours
$R1 \to H1 \to H2 \to G4$	29
$R1 \to H2 \to G4$	32
$R1 \to H1 \to G4$	38

LP optimal length 29 h on $R1 \to H1 \to H2 \to G4$, matching a from-scratch Dijkstra implementation exactly.

The cheap inter-terminal arc (5 h) beats both direct terminal routes. The LP produced a 0-1 flow without being told to.

Maximum flow: a capacity question, not a cost question

Construction

Add a super source SRC with an arc to each plant of capacity equal to its supply, and a super sink SNK with an arc from each gigafactory of capacity equal to its demand.

$$\max \sum_j t_j \quad \text{s.t.} \quad \text{inflow}(n) + s_n = \text{outflow}(n) + t_n,$$

$$0 \le f_{ij} \le u_{ij}, \quad 0 \le s_i \le \text{supply}_i, \quad 0 \le t_j \le d_j.$$

The scenario

Both terminals are under berth maintenance, which **halves the capacity of every terminal-to-gigafactory arc**. How much can the network still deliver?

Result

Maximum weekly throughput 1,100 t against 1,130 t of demand.

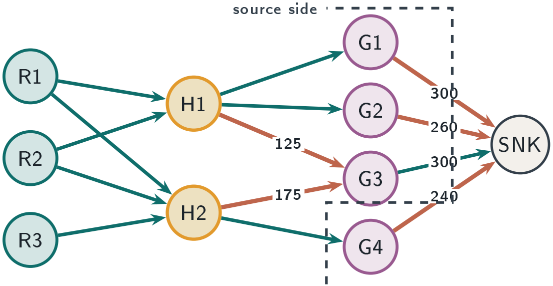
t/week	G1	G2	G3	G4
demand	300	260	330	240
delivered	300	260	300	240

The entire 30 t shortfall lands on **G3**, and only on G3.

Max-flow min-cut

The maximum flow equals the minimum capacity of a cut separating SRC from SNK: LP duality specialized to a network. The cut is read off the capacity duals.

The minimum cut, and the only investment that helps



Rust arcs form the minimum cut. Its capacity is $125 + 175 + 300 + 260 + 240 = 1{,}100$ t, equal to the maximum flow, confirmed both from the LP duals and by enumerating all 512 cuts of the nine-node network.

The engineering reading, and it is precise

Adding capacity **anywhere except on an arc of the cut** changes nothing at all. To recover the missing 30 t you must widen $H1 \to G3$ or $H2 \to G3$. No other investment will do it.

Why this is the useful output

A debottlenecking study that reports only the throughput number has thrown away the answer. The cut is the list of assets worth money, and everything not in the cut is worth exactly zero at the margin.

Total unimodularity: integers for free

Definition and theorem

A matrix E is totally unimodular (TU) if every square submatrix has determinant 0, $+1$ or -1. **The node-arc incidence matrix of a directed graph is TU.**

Theorem (Hoffman and Kruskal). If E is TU and b, u are integral, then every basic feasible solution of $\{f : Ef = b,\ 0 \le f \le u\}$ is integral. The simplex method returns a basic solution, so it returns an integer answer.

Why

A basic solution solves $Bf_B = b'$ for a nonsingular square submatrix B of E. By Cramer's rule $f_B = B^{-1}b'$ with $\det B = \pm 1$, so B^{-1} has integer entries, and an integer matrix times an integer vector is an integer vector.

Proof sketch for TU, by induction on submatrix size: a square submatrix either has a zero column, or a column with one non-zero entry (expand along it), or every column has both a $+1$ and a -1, in which case the rows sum to zero.

Practical significance

- Transportation, transshipment, shortest path, maximum flow and assignment never need integer variables. Solve the LP, get integers, at LP speed.
- TU is a property of the **constraint matrix**, not of the costs. Changing the costs cannot make the answer fractional.
- Every flow reported so far came out integral from a pure LP solve.

TU is fragile: one innocuous side constraint

The added requirement

Terminal $H1$ may handle at most 35 percent of everything that leaves the plants:

$$\sum_{(i,H1)\in A} f_{i,H1} \;\le\; 0.35 \sum_{i\in S} \sum_{(i,j)\in A} f_{ij}.$$

This row is not part of the incidence structure: it has coefficients 1 and -0.35 spread across arcs that share no node.

What happens

Cost rises from 25,850.00 to 26,195.00 USD/week, and four arcs come out at exactly one half tonne: $f_{R1,H1} = 365.5$, $f_{R1,H2} = 84.5$, $f_{H1,G2} = 95.5$, $f_{H2,G2} = 164.5$.

What destroys total unimodularity

- A percentage-share or blending-ratio constraint across arcs.
- A fixed-charge binary linked to a flow.
- A joint capacity shared by arcs that do not meet at a node.
- Any row whose coefficients are not $\{0, +1, -1\}$ in the incidence pattern.

The modeling consequence

If half a tonne is physically meaningless you now need integer variables, and the model changes class from LP to MILP: Week 5 for the formulation, Week 11 for the algorithm. Know before adding a row whether you are about to pay that price.

Break, 15 minutes

Next: open `excel/W03_transportation_STUDENT.xlsx` and have OpenSolver loaded.

Activity 2 starts at 1:45. You will reproduce a stated optimum, then break the supply-demand balance on purpose.

Activity 2 | 25 min | computer, individual: transportation in OpenSolver

Part 1, reproduce the optimum

Open `excel/W03_transportation_STUDENT.xlsx`, sheet `Transportation`. Complete the yellow cells: the row sums `G12:G14`, the column sums `C15:F15`, and the objective `C19` as one `SUMPRODUCT` over the whole matrix. Run OpenSolver with CBC on objective `$C$19`, To: Min, changing cells `$C$12:$F$14`, constraints `$G$12:$G$14 <= $I$12:$I$14` and `$C$15:$F$15 >= $C$17:$F$17`.

The objective cell must read 23,570.00 USD/week. If it does not, you have a formula error, and that is the point of this part.

Part 2, break the balance on purpose

1. Reduce total supply below total demand (for example set $R2$ to 200 t). Re-solve. Record exactly what the solver reports.
2. Add a **dummy source** with a large penalty cost on every lane, give it the missing tonnage, and re-solve. Record the new objective and which gigafactory the dummy serves.

Grouping individual. **Time** 25 min. **Hand back** nothing: the instructor walks the room and checks screens.

Activity 2 | [ANSWER] | what the dummy source really is

Part 1

Objective `C19` $= 23{,}570.00$ USD/week, identical to the Pyomo model in `W03_modeling2_networks.ipynb`. Plan: R1 ships 300 to G1 and 150 to G2; R2 ships 110 to G2 and 270 to G3; R3 ships 60 to G3 and 240 to G4.

Sensitivity report: supply duals $u = (0, -5, -2)$, demand duals $v = (18, 24, 27, 23)$ USD/t. Dual objective $= 23{,}570.00$, so strong duality holds. Every shipment is a whole number of tonnes from a pure LP.

Part 2

Supply below demand. With $\sum_i s_i < \sum_j d_j$ and demand written with $\geq$, the model is **infeasible**. OpenSolver reports infeasibility; Excel Solver reports "Solver could not find a feasible solution". No plan exists, because the model has been told that every tonne of demand must be served.

Dummy source. Adding a fictitious plant whose shipping cost is a penalty M restores feasibility. The dummy's shipments are the tonnes *not* served.

The question to the room

What does the dummy cost represent in the real system? **The cost of not serving a customer**, previously implicit and now written down. Choosing M is a business decision: contractual penalty, lost margin, or the price of an emergency purchase. The dummy also reveals which customer is cheapest to disappoint, which is often more useful than the cost itself.

From places to units: the process superstructure

The reinterpretation

Same mathematics, different meaning of the objects: **nodes are process units**, **arcs are material streams**. It is the standard way to pose a process *synthesis* question, that is, "which flowsheet should we build", rather than an operating question, "how should we run the flowsheet we have".

The idea

Build one network that contains **every candidate route at once**, and let the optimizer decide how much material goes down each route. A route that is not worth using simply receives zero flow.

Nothing here needs a binary variable

The split of a stream between competing units is a *continuous* variable bounded below by zero, and the optimizer drives the uninteresting splits to zero by itself.

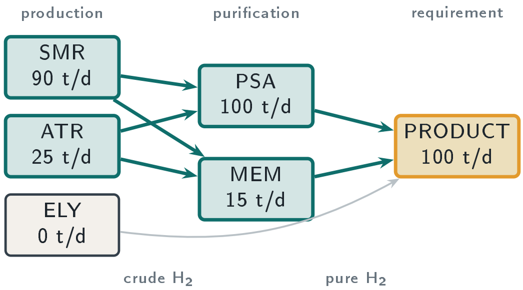
The Week 2 vocabulary covers it

- index set over units and streams,
- balance around every unit and every mixing or splitting node,
- capacity constraint on every unit,
- linking constraint wherever two routes share a downstream unit.

The hydrogen superstructure: verbal problem and flowsheet

A battery-materials site needs 100 t/day of purified hydrogen for cathode-precursor reduction. Three production routes exist: steam methane reforming (`SMR`), autothermal reforming (`ATR`) and water electrolysis (`ELY`). `SMR` and `ATR` produce a crude stream that must be purified; `ELY` produces hydrogen already pure enough. Two purifiers are available: pressure swing adsorption (`PSA`), which recovers more but costs more to run, and a membrane (`MEM`), cheaper and recovering less. Crude from either reformer may go to either purifier in any proportion.

Unit	$F_u^{\max}$ t/d	k_u USD/t	η_u
SMR	90	1300	
ATR	70	1550	
ELY	40	3400	
PSA	100	200	0.88
MEM	80	100	0.80



Every candidate route drawn at once. Throughputs shown are the LP optimum. The `ELY` route is present in the model and receives zero flow, which is the answer, not an omission.

The algebraic model, and why there is no split fraction

Linear superstructure

min	$\sum_{u \in U} k_u F_u$	[USD/day]
s.t.	$F_{\mathrm{SMR}} + F_{\mathrm{ATR}} = F_{\mathrm{PSA}} + F_{\mathrm{MEM}}$	crude balance [t/day]
	$\eta_{\mathrm{PSA}} F_{\mathrm{PSA}} + \eta_{\mathrm{MEM}} F_{\mathrm{MEM}} + F_{\mathrm{ELY}} = D$	product balance [t/day]
	$0 \le F_u \le F_u^{\max} \quad \forall u \in U$	capacity

The cheapest lesson of superstructure modeling

There is no split fraction in this formulation and there does not need to be one. The split of crude between PSA and MEM is $F_{\mathrm{PSA}}/(F_{\mathrm{SMR}} + F_{\mathrm{ATR}})$, and it is **recovered after the solve, not imposed before it**. Writing splits as ratios introduces a division and makes the model nonlinear for no gain.

The two balances are the linking constraints

The crude balance links the reformer block to the purifier block; the product balance links all three routes to the single requirement. Delete them and the model separates into five unrelated one-variable problems with no feasible answer.

The linear superstructure selects routes with no binary variable

Cost of one tonne of *purified* hydrogen, by route

Route	USD per t pure
SMR → PSA	1704.55
SMR → MEM	1750.00
ATR → PSA	1988.64
ATR → MEM	2062.50
ELY (no purification)	3400.00

Computed by hand as $(k_{\text{prod}} + k_{\text{pur}})/\eta_{\text{pur}}$, so that the LP answer can be predicted before it is produced.

LP optimum

Minimum operating cost = 177,250.00 USD/day, that is 1,772.50 USD per tonne of purified hydrogen.

SMR runs full at 90 t/day; all of that plus 10 t/day of ATR crude goes through PSA at its full 100 t/day; the remaining 15 t/day of ATR crude goes through MEM; ELY stays at zero. The recovered split of crude is 100/115 to PSA and 15/115 to MEM.

The dual of the product balance is 2,062.50 USD/t

Exactly the cost of the ATR → MEM route, the most expensive route actually carrying flow. The marginal tonne of product is made by the marginal route: a Week 2 shadow price transferred to a flowsheet.

Where binaries would enter, and what they cost

Unit selection: the fixed-charge pair

$y_u = 1$ if unit u is built, 0 otherwise.

$$\min \sum_u k_u F_u + \sum_u f_u y_u \quad \text{s.t.} \quad F_u \le F_u^{\max} y_u,$$

plus the balances and bounds as before,
$y_u \in \{0, 1\}$.

Equivalent daily capital charge f_u (USD/day): SMR 22,000, ATR 15,000, ELY 9,000, PSA 12,000, MEM 30,000.

Capital charges change the flowsheet, not just the cost

MILP optimum	248,720.00 USD/day
LP relaxation of it	222,232.14 USD/day
integrality gap	26,487.86 (10.65 %)

The MILP builds **SMR, PSA and ELY only**. ATR and the membrane are dropped, and electrolysis, which the LP refused to use at all, is switched on to make up the 20.8 t/day that SMR and PSA cannot cover.

The point of process synthesis

A route with a high running cost can still be correct if it avoids a large capital commitment. Formulation strength is Week 5; branch and bound is Week 11.

One physically motivated nonlinearity, and the class changes again

Load-dependent recovery

A real adsorber or membrane recovers a larger *fraction* when run below design, because contact time per tonne is longer. A first-order representation:

$$\eta_u(F_u) = \eta_u^0 - s_u F_u, \qquad \eta_u F_u = \eta_u^0 F_u - s_u F_u^2.$$

Coefficients chosen so the design values are reproduced exactly: $0.92 - 0.0004(100) = 0.88$ for PSA and $0.84 - 0.0005(80) = 0.80$ for MEM.

Consequence

The product balance becomes a quadratic *equality*, so the feasible set is not convex. Without binaries it is an NLP with no global guarantee; with them it is a nonconvex MINLP, the hardest class in this course.

MINLP optimum = 247,496.00 USD/day, marginally below the MILP value, because PSA running at 90 t/day rather than its design 100 recovers 0.884 instead of 0.880 and therefore needs slightly less electrolysis behind it.

Verification is not optional

A nonconvex MINLP solver can return a local solution and report it as optimal. The notebook therefore enumerates all 32 selection patterns and solves the NLP behind each. Week 4 is where this becomes the whole lesson.

Activity 3 | 20 min | groups of 3 to 4: sketch a superstructure

The problem

A site must deliver 100 t/day of purified hydrogen. **Three production routes** are candidates: steam methane reforming, autothermal reforming and water electrolysis. The first two produce a crude stream; electrolysis produces hydrogen that is already pure enough. **Two purification technologies** are candidates for the crude stream: pressure swing adsorption and a membrane unit. Crude from either reformer may go to either purifier in any proportion.

Your task, on one sheet

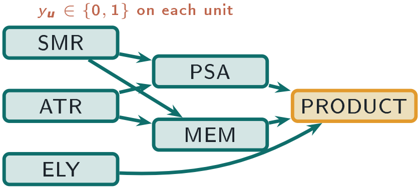
1. Draw the superstructure: every unit, every candidate stream, the mixing and splitting nodes, and the product node.
2. Mark every decision as **continuous** (how much through each route) or **discrete** (whether to build a unit at all), using two clearly different symbols.
3. State which constraints **change** if unit selection is added, and write the one new constraint family in algebra.

Logistics

Grouping groups of 3 to 4.
Time 20 min; two groups present at the board at 15 min.
Hand back one sketch per group, with the decision types marked and the new constraint family written out.

Do not compute anything. The deliverable is the structure and the classification of the decisions.

Activity 3 | [ANSWER] | continuous splits, and the one row that adds a binary



5 units, 7 candidate streams, 2 balances.

Which decisions are which

Continuous: the throughput $F_u \geq 0$ of every unit, and therefore every split of crude between PSA and MEM. Splits are *outputs* of the solve, not inputs.

Discrete: only "is unit u built at all". Nothing else in this problem is discrete.

What changes when selection is added

- **New:** $F_u \leq F_u^{\max} y_u$, $y_u \in \{0, 1\}$, one row per unit.
- **Objective:** gains $\sum_u f_u y_u$, a cost that does not scale with throughput.
- **Unchanged:** both balances and the physical bounds. The plain capacity row $F_u \leq F_u^{\max}$ becomes redundant, absorbed by the new row.

Closing

Revisited twice: Week 5 formulates the discrete version and asks how tight the link should be; Week 11 shows how branch and bound closes the resulting gap, which is 10.65 percent on this instance.

Summary

Concepts

- One template covers the family: choose the graph, the costs, the capacities and b, and transportation, transshipment, shortest path and maximum flow all follow.
- The node balance is the model, and it is the same conservation law as a material balance around a unit.
- Network LPs return integers for free by total unimodularity, and the property belongs to the matrix, not to the costs.
- TU is fragile: one share constraint or one fixed-charge binary and fractional vertices reappear.
- Duals carry the engineering content: node potentials price material by location, and the minimum cut is the only place extra capacity is worth anything.
- A superstructure is a network whose nodes are units. Route choice is continuous; only "build or not" is discrete.

Numbers established today

Transportation optimum	23,570.00 USD/week
Transshipment optimum	25,850.00 USD/week
Shortest path $R1 \to G4$	29 h
Maximum flow / demand	1,100 / 1,130 t
Minimum cut capacity	1,100 t, 5 arcs
With a 35 % share limit	26,195.00 USD/week
Superstructure LP	177,250.00 USD/day
Product balance dual	2,062.50 USD/t
Superstructure MILP	248,720.00 USD/day
Its LP relaxation	222,232.14 USD/day
Integrality gap	26,487.86 (10.65 %)
Superstructure MINLP	247,496.00 USD/day

Coming next

Week 4: blending, pooling and quality specifications, and the point where a spreadsheet stops being able to tell you the truth.

Before next week

Reading

- Williams, *Model Building in Mathematical Programming*, Ch. 5 (network models).
- Edgar, Himmelblau and Lasdon, Ch. 7, sections on transportation and network flow.
- Rao, Ch. 3, the transportation problem as a special-structure LP.

Notebook to finish

`notebooks/W03_modeling2_networks.ipynb`, all sections, plus **Exercises 1, 2 and 4**. Exercise 4 re-verifies max-flow min-cut on a modified instance and is the one that repays the effort.

Homework

HW1 is due at the start of Week 4. It covers linear and network model construction, that is Weeks 2 and 3 together. Bring the transportation workbook with you: one question asks you to compare the sheet and the Pyomo route on the same instance.

Install before Week 4

Week 4 crosses from the spreadsheet to Python and needs a nonlinear solver. Install `ipopt` now with `conda install -c conda-forge ipopt`, or run the IDAES extensions installer. Test it before the session with `SolverFactory('ipopt').available()`.

Further reading

- **Williams**, *Model Building in Mathematical Programming*, 5th ed., Ch. 5 and Ch. 6: network formulations, and the conditions under which a formulation retains integrality.
- **Ahuja, Magnanti and Orlin**, *Network Flows: Theory, Algorithms and Applications*. The standard reference for the specialized algorithms (network simplex, preflow-push) that beat general LP on these problems by orders of magnitude.
- **Edgar, Himmelblau and Lasdon**, *Optimization of Chemical Processes*, 2nd ed., Ch. 7 and Ch. 11: network models and flowsheet optimization in process applications.
- **Biegler, Grossmann and Westerberg**, *Systematic Methods of Chemical Process Design*, for superstructure representation and process synthesis.
- **Rao**, *Engineering Optimization*, 4th ed., Ch. 3, on special-structure linear programs.

Forward references made today: total unimodularity breaks in Week 5 (binaries) and the resulting search is Week 11; degenerate duals and shadow prices are Week 8; the nonconvexity that appeared in the last superstructure is the whole of Week 4.