

---

SPEA-V348 Management Science · Week 10 review

# **Networks of supply, and whole units**

Both sessions, every slide, fully revealed. Read it without me in the room.

Mon Oct 26 and Wed Oct 28, 2026

---

# The week at a glance

---

|                   |                                                                                                                                                     |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Mon Oct 26</b> | Transportation grid and rounding check: 30-minute opening, 35-minute TA-facilitated practice, photographs, then the instructor's eight-minute check |
| <b>Wed Oct 28</b> | Larger grids and integer-model extensions, the company's hub and flow balance, then assignment and the take-home briefing                           |
| <b>Reading</b>    | Taylor 13e, Chapter 6, pp. 261 to 279, and Chapter 5, pp. 208 to 220, skipping QM                                                                   |
| <b>Due</b>        | The Week 10 workbook, on Canvas. Tier 1: AI as a tool, with disclosure                                                                              |

---

ON THE SITE · the Week 10 page

The three widgets, the two self-checks, and the workbook download.

[v348.jima.me/week/w10/](https://v348.jima.me/week/w10/)



# Session 1

Mon Oct 26

# One company, two plants, two stores

*The question this session answers*

---

## PUZZLE

Plant A can ship **120** units and Plant B **80**. Store 1 needs **100** and Store 2 needs **100**, so supply and demand both total 200. Shipping costs per unit:

| \$ per unit   | Store 1 | Store 2 | Supply |
|---------------|---------|---------|--------|
| Plant A       | 8       | 6       | 120    |
| Plant B       | 5       | 9       | 80     |
| <b>Demand</b> | 100     | 100     | 200    |

*What shipping plan meets every store's demand at the **minimum total cost**?*

## Where we are

*Two blocks today, in this order*

---

NOW

The grid and its cheapest plan



NEXT

Whole vehicles and the rounding check

*First read the four routes, write the grid and find its cheapest plan. Then the same company buys vehicles against a budget and driver limit: half a van is not allowed, and rounding can fail.*

## Build a feasible shipment grid first

---

| Route        | Units                | Cost per unit | Route cost    |
|--------------|----------------------|---------------|---------------|
| A to Store 1 | 60, our trial choice | 8             | $60(8) = 480$ |
| A to Store 2 | $120 - 60 = 60$      | 6             | $60(6) = 360$ |
| B to Store 1 | $100 - 60 = 40$      | 5             | $40(5) = 200$ |
| B to Store 2 | $80 - 40 = 40$       | 9             | $40(9) = 360$ |

Check the supplies: A sends 120, B sends 80. Check the demands: each store receives 100. Cost is  $480 + 360 + 200 + 360 = 1,400$  dollars.

*The plan works. Now let Solver change the shipment quantities while preserving those four totals.*

Week 10 page, opening trial plan.

## The grid, written out

*One equality per row, one per column, and shipments cannot go negative*

---

| Ingredient         | In symbols                                       | From the brief                   |
|--------------------|--------------------------------------------------|----------------------------------|
| Decision variables | $x_{A1}, x_{A2}, x_{B1}, x_{B2}$                 | units on each of the four routes |
| Objective          | minimize $8x_{A1} + 6x_{A2} + 5x_{B1} + 9x_{B2}$ | cost grid times shipment grid    |
| Supply rows        | $x_{A1} + x_{A2} = 120; x_{B1} + x_{B2} = 80$    | everything each plant ships      |
| Demand columns     | $x_{A1} + x_{B1} = 100; x_{A2} + x_{B2} = 100$   | everything each store receives   |

*In Solver the objective cell is one SUMPRODUCT of the two grids, and each equality is one constraint row.*

# The transportation model, in symbols

*Three lines, and they are exactly the cells you type into Solver*

---

$$\text{minimize } \sum_i \sum_j c_{ij} x_{ij}$$

**The decision variables:** units sent from source  $i$  to destination  $j$ , one per route. Four of them here

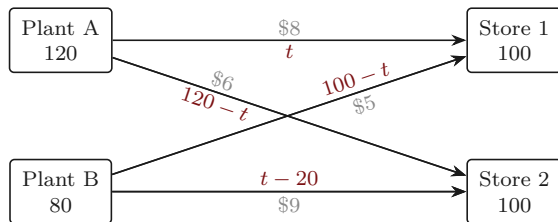
**The unit cost** on that route: \$8, \$6, \$5, \$9. Cost times units, added over every route

**Subject to:** each source ships out its whole supply,  $\sum_j x_{ij} = s_i$ ; each destination receives its whole demand,  $\sum_i x_{ij} = d_j$

Taylor 13e, pp. 261 to 263. Balanced means total supply equals total demand, which is what lets both families be equalities.

## A balanced 2 by 2 has one free number

*Choose the units on Plant A to Store 1 and everything else is forced*



**Row A:** A to Store 2 is  $120 - t$ . **Column 1:** B to Store 1 is  $100 - t$

**Row B:** B to Store 2 is  $80 - (100 - t) = t - 20$ . This must be at least 0

**Cost:**  $6t + 1,040$ , rising \$6 for each extra unit on A to Store 1

*The floor is  $t = 20$ : any lower and B to Store 2 goes negative. Ship 20, 100, 80, 0 for **\$1,160**: the **cheap** \$5 route flat out, the **dear** \$9 route empty.*

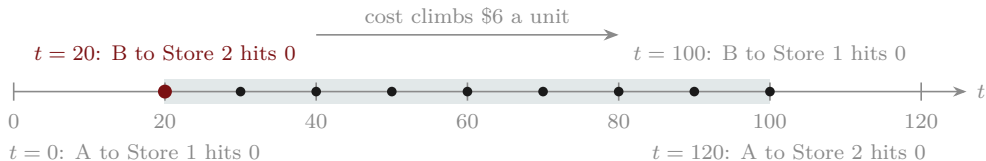
$$8t + 6(120 - t) + 5(100 - t) + 9(t - 20) = 6t + 1,040; \text{ at } t = 20, 160 + 600 + 400 + 0 = 1,160.$$

## Whole units, for free

*The 2 by 2 as a linear program: one variable, a legal range, and the cheapest end of it*

---

Look at the plans again. Every shipment came out a **whole number**, and nobody asked for that.



Both ends are whole because they came from whole supplies and demands ( $100 - 80 = 20$ , and 100), and the best plan sits at an end. The structure did that, and it is why Solver dispatches large grids in a blink.

*The structure handed you integers. Now take the structure away.*

A property of the transportation structure Taylor does not name; his Chapter 6 answers all show it.

## Where we are

*The grid is checked; now require whole vehicles*

---

DONE

✓ The grid and its cheapest plan



NOW

Whole vehicles and the rounding check

*The 2 by 2 costs \$1,160 and its shipments are whole without an integer restriction. Now the company plans to expand and buy vehicles: a budget and driver limit do not give that same guarantee.*

## Trucks and vans: the company cannot buy half a van

*Maximize monthly capacity in thousands of units, with the budget counted in \$10,000 units*

---

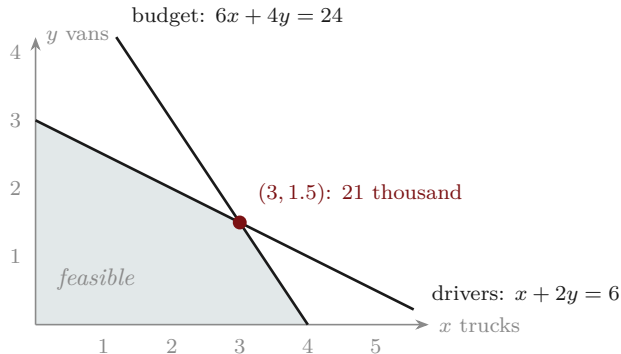
Plant C and Store 3 need trucks. A **box truck** costs \$60,000, needs 1 driver, and moves 5,000 units a month. A **cargo van** costs \$40,000 and runs two shifts, so it needs 2 drivers, and moves 4,000. There is \$240,000 and there are 6 drivers.

maximize  $5x+4y$    subject to    $6x+4y \leq 24$  (budget),  $x+2y \leq 6$  (drivers),  $x, y \geq 0$

*Two variables and two limits, with no rows and columns behind them. First allow fractions: draw the two lines, shade what survives both, and go to a corner.*

# The best corner is not a whole number

*Allow fractions to find an upper bound on capacity*



**The budget line**  $6x + 4y = 24$ :  
four trucks spend it all,  $(4, 0)$

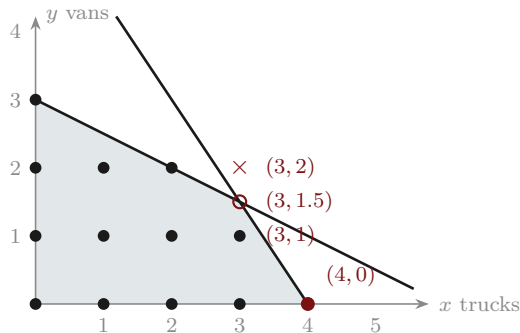
**The driver line**  $x + 2y = 6$ : three  
vans use every driver,  $(0, 3)$

**The feasible region**: every plan  
inside it fits both limits

**The best corner**, where they  
cross:  $(3, 1.5)$  moves  $15 + 6 = 21$   
thousand

*Half a van does not exist. Solving without the whole-number rule is the **LP relaxation**: a **ceiling** no whole-number plan can beat.*

## Round it?



**Round up** to  $(3, 2)$ : budget 26 of 24 and 7 drivers of 6. Off the picture, not allowed

**Round down** to  $(3, 1)$ : budget 22, drivers 5, moving 19,000. Legal, and not the best dot

**The best dot** is  $(4, 0)$ : budget 24 exactly, 4 drivers, moving 20,000

*The best whole-number plan was not next to the fractional one and it was not a rounding of it. It was a **different mix**: four trucks, no vans, the budget spent exactly.*

## Back to the two plants

---

- Ship **20** on Plant A to Store 1, so 100 on A to Store 2, 80 on B to Store 1, and nothing on B to Store 2: **\$1,160**, the lowest the slider can reach.
- The intuition held: the cheap B to Store 1 route runs flat out, the dear B to Store 2 route carries nothing, and Plant A quietly covers what is left.

*Read the grid, write the equalities, minimize the cost sum. Wednesday adds Plant C and Store 3, so the same moves need Solver.*

# Wednesday

---

a grid

sources down,  
destinations across

$= s_i, = d_j$

one equality per row and  
column

\$1,160

the cheap route flat out

(4, 0)

not a rounding of (3, 1.5)

*Wednesday extends the grid, adds integer-model types, and asks the same company two new questions: whether a **hub** between the plants and the stores, where whatever flows in must flow out, can beat \$1,160, and which crew should take each store's run, a grid where every supply and demand is 1: **matching**.*

Week 10 take-home due Sunday, November 1, at 11:59 PM. Full briefing on Wednesday.

## Group practice

---

### IN CLASS

A campus coffee co-op trucks beans from two roasteries (50 kg and 40 kg) to three cafes wanting 30 kg each, with a cost per kilogram on every route. Fill the cost matrix, find a low-cost plan by hand that honors every row and column total, and cost it. Then a purchase that cannot be fractional: vans and cargo e-bikes, where the relaxation says 4.5 vans. Round up, round down, then find the best whole-number plan. **Every member's name goes on the sheet**; one of you photographs it and **uploads the photo to Canvas** before leaving. One participation point, half today's attendance record.

ON THE SITE · Week 10 practice sheet

The same sheet is on the site if you want it on a screen.

[v348.jima.me/assets/docx/v348-w10-s1-group-practice.docx](https://v348.jima.me/assets/docx/v348-w10-s1-group-practice.docx)



## Photograph your attempt, then check it together

*Two minutes to capture your work; eight for the instructor's demonstration*

---

### IN CLASS

Check every member's name and photograph your group's own attempt before the solution is shown. One member uploads the photo to Canvas by the end of class.

*We close by checking the method, the main results with units, and a common mistake. Participation credit is for attempting the work.*

# Session 2

Wed Oct 28

## Same company, expanded questions

*The question this session answers*

---

### PUZZLE

The company opens a **hub** between Plants A and B and Stores 1 and 2. A unit costs **\$2** to truck into it from either plant, then **\$4** on to Store 1 or **\$3** on to Store 2, and Monday's four direct routes stay open. Each store's delivery run also needs a crew: Crew 1 takes **7** hours on the Store 1 run and **9** on the Store 2 run; Crew 2 takes **8** and **5**. One crew per run.

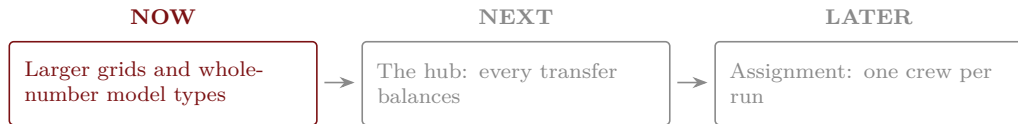
Monday's 200 units went direct, for \$1,160. Today they get a second way to travel, and the stores get people: a grid where every number is 1.

*Direct, through the hub, or some of each: what does the best plan cost? And which crew should take which run?*

## Where we are

*Three blocks today, in this order*

---



*First extend Monday's grid and whole-number model. Then add a hub and its balance equation. Finally match the crews to runs, from two matchings to a larger grid where greedy can lose.*

## Now scale it past what you can eyeball

*Plant C and Store 3 open: three plants, three stores, \$ per unit, balanced at 300*

---

| \$ per unit | Store 1 (100) | Store 2 (100) | Store 3 (100) | Supply |
|-------------|---------------|---------------|---------------|--------|
| Plant A     | 8             | 6: 20         | 4: 100        | 120    |
| Plant B     | 5: 80         | 9             | 7             | 80     |
| Plant C     | 3: 20         | 4: 80         | 5             | 100    |

The cheapest cells alone cannot work: Plant A would send 120 to a store that wants 100, Plants B and C would both pile onto Store 1, 180 units for a store that wants 100, and Store 2 would get nothing. Where the leftovers go is the whole problem. Solver's plan is above, at **\$1,300**.

*Same skill as the 2 by 2, read the grid and write the equalities, but too many combinations to guess. **That is what Solver is for.***

# The lattice, on your laptop

---

## IN CLASS

Open the second Session 1 widget on the Week 10 page: the same picture, live, every dot a whole-number plan inside both the budget and the driver limit. Press **LP optimum** to see  $(3, 1.5)$ , then **Round it** and read which limits the rounded plan breaks, then **Integer optimum** and watch it land on  $(4, 0)$ .

ON THE SITE · the whole-number widget

The lattice of legal plans, with the three candidates worked out under it.

[v348.jima.me/week/w10/](http://v348.jima.me/week/w10/)



## Three kinds of integer model, and one line in Solver

---

### Total integer

every variable whole: how many trucks, how many vans

### 0–1

every variable a *binary* yes or no: open Plant C, or not

### Mixed

some whole, some continuous: open Plant C or not, then ship any amount

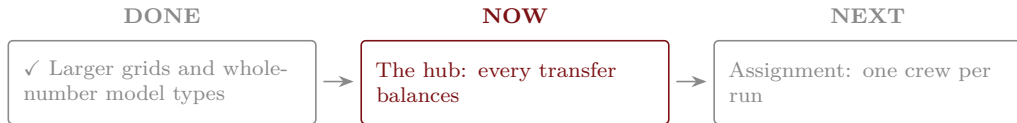
In Solver the change is one constraint on the decision cells whose relation is `int`, or `bin` for yes or no. What it costs is speed: Solver can no longer slide along edges to a corner, and searches the dots by **branch and bound**, solving relaxed programs and discarding whole branches that cannot beat the best whole-number plan found so far.

Taylor 13e, pp. 208 to 210 for the three models, and pp. 213 to 220 for the Solver setup.

## Where we are

*Larger models are set up; now add a transfer node*

---

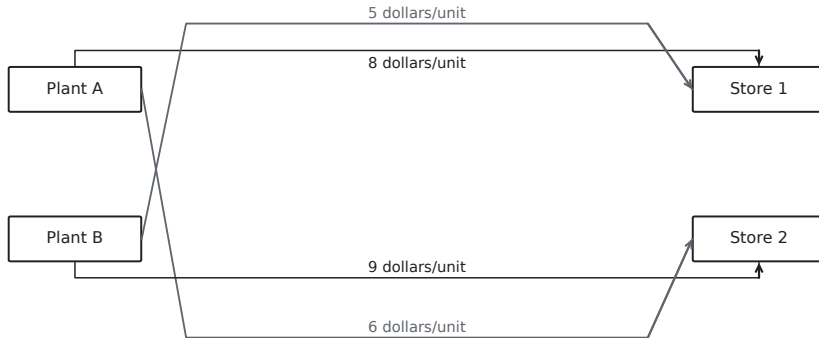


*The larger transportation grid still uses row and column totals. Vehicles need explicit whole-number restrictions. Return to Monday's two plants and two stores and give their goods a second route, through a hub.*

# Build the network: keep the direct routes

*Unit costs stay attached to their routes*

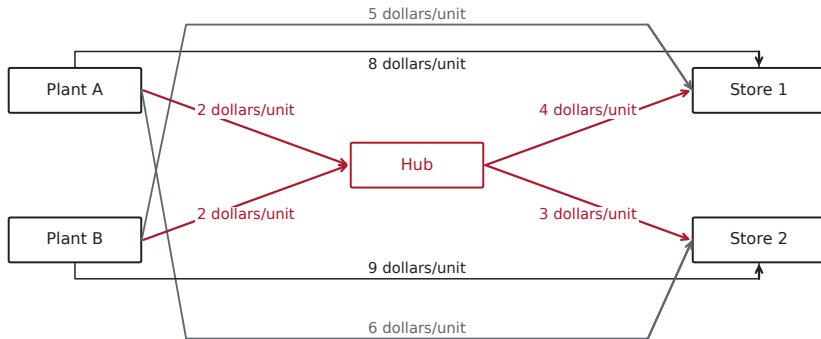
---



Week 10 page. Supply: Plant A 120, Plant B 80. Demand: 100 at each store.

## Build the network: add the hub

*Red adds four optional legs; it does not replace the direct routes*



Week 10 page. Supply: Plant A 120, Plant B 80. Demand: 100 at each store.

## Price the all-hub trial on its four legs

---

| Leg            | Units | Cost per unit | Dollars |
|----------------|-------|---------------|---------|
| A to Hub       | 120   | 2             | 240     |
| B to Hub       | 80    | 2             | 160     |
| Hub to Store 1 | 100   | 4             | 400     |
| Hub to Store 2 | 100   | 3             | 300     |

*Total: 1,100 dollars. The hub receives  $120 + 80 = 200$  and dispatches  $100 + 100 = 200$ . Now check whether a mix of direct and hub routes can do better.*

## Both open: each unit takes the cheaper path

*The hub has no capacity limit, so every plant and store pair compares two prices*

---

| \$ per unit        | Direct | Through the hub | Cheaper |
|--------------------|--------|-----------------|---------|
| Plant A to Store 1 | 8      | $2 + 4 = 6$     | hub     |
| Plant A to Store 2 | 6      | $2 + 3 = 5$     | hub     |
| Plant B to Store 1 | 5      | $2 + 4 = 6$     | direct  |
| Plant B to Store 2 | 9      | $2 + 3 = 5$     | hub     |

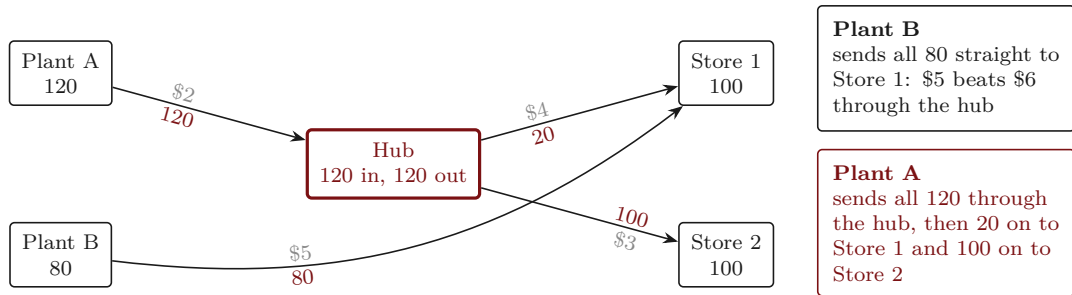
That is Monday's grid with the cheaper price on each pair, 6, 5, 5 and 5, and the two-plant grid's free number finishes it. With  $t$  units from Plant A to Store 1 the cost is  $6t + 5(120 - t) + 5(100 - t) + 5(t - 20) = t + 1,000$ , still lowest at  $t = 20$ .

*Plant B skips the hub for Store 1, where \$5 direct beats \$6 through it. Every other unit is cheaper through the hub.*

## The best plan uses both

*Plant B goes direct, Plant A goes through the hub, and the other routes carry nothing*

---



$80(5) + 120(2) + 20(4) + 100(3) = \textbf{\$1,020}$ , against \$1,160 direct only and \$1,100 through the hub only. A hub is an **option**, not a forced detour.

## Flow balance, in symbols

*The one new constraint a hub adds, on the plan that uses both*

---

$$\text{inflow} = \text{outflow}$$

**Inflow**, everything arriving at the hub:  $120 + 0 = 120$  units from Plants A and B

**Outflow**, everything leaving it:  $20 + 100 = 120$  units to Stores 1 and 2

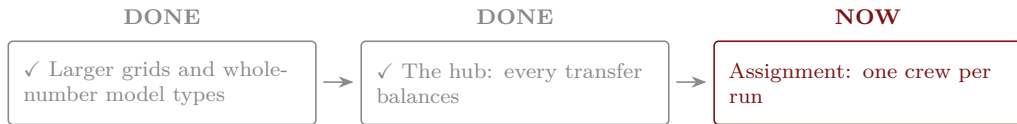
**Equal**, because a pure transfer node keeps nothing. In Solver: one cell, arriving minus leaving, set to 0

Leave the cell out and the hub can hand out goods it never received: write the plants' rows as  $\leq$  and Solver ships all 200 units out of the hub and none in, a \$700 plan for goods no plant made. The take-home's cross-dock asks you to write this cell yourself.

## Where we are

*The hub balances; now match crews to runs*

---



*A hub adds one equation, everything in equals everything out: with both routes open, 120 in and  $20 + 100$  out, for \$1,020. The crews have no hub. Every supply and every demand is 1, and the block asks what that does to Monday's grid.*

## Assignment: a transportation grid where everything equals 1

---

**Every supply is 1**

each crew is a source with one unit to give: itself

**Every demand is 1**

each run is a destination that needs exactly one crew

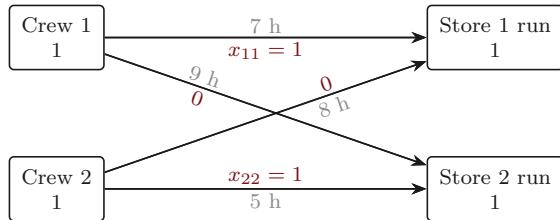
**Every variable is 0 or 1**

$x_{ij} = 1$  if crew  $i$  takes store  $j$ 's run. Rows sum to 1, columns sum to 1

*You are not splitting shipments. You are **matching**: each crew to one run, the same model that puts each machine on one task or each official on one game. Solver returns clean 0s and 1s without an `int` constraint, for Monday's reason.*

## Both matchings, costed

*Monday's 2 by 2 picture, with hours on the routes and one unit at every end*



**Straight:** Crew 1 on the Store 1 run and Crew 2 on the Store 2 run,  $7+5 = 12$  hours

**Crossed:** Crew 1 on the Store 2 run and Crew 2 on the Store 1 run,  $9+8 = 17$  hours

**One free number:** set  $x_{11} = 1$  and the other three cells are forced, just like Monday's  $t$

*Two crews give two matchings, and **12 hours** is optimal. The interesting case is the one you cannot enumerate.*

## From two crews to four

*The assignment grid grows*

---

Two crews have two matchings. Four crews have  $4! = 24$ .

*Rows and columns summing to 1 leave one free number, and  $x_{11} = 1$  costs 12 hours against 17. The company's other two crews and two more runs make 24 matchings, too many to cost by hand, and this block asks whether a shortcut can stand in for the model.*

## Why you cannot eyeball the big ones

*The company's other two crews join, with Store 3's run and the hub run*

---

| Hours  | Store 1 run | Store 2 run | Store 3 run | Hub run  |
|--------|-------------|-------------|-------------|----------|
| Crew 1 | <b>7</b>    | 9           | 5           | 3        |
| Crew 2 | 8           | <b>5</b>    | 4           | 8        |
| Crew 3 | 12          | 12          | 7           | <b>5</b> |
| Crew 4 | 12          | 11          | <b>5</b>    | 8        |

Greedy: Crew 1 on the hub run (3), then Crew 2 on Store 3's run (4), then Crew 4 on Store 2's (11), and Crew 3 is left with Store 1's (12): **30 hours**. The optimum, in bold, never touches the 3:  $7 + 5 + 5 + 5 = \mathbf{22}$ , eight hours less, and the puzzle's own matching is still inside it.

*The cheapest single choice can poison the rest of the grid. Assignment optimizes the **whole matching** at once.*

## How many matchings are there?

---

2

two crews, two runs

$4! = 24$

four crews, four runs

$10! = 3,628,800$

ten crews, ten runs

Nobody checks 3.6 million by hand, and the greedy shortcut gets worse as the grid grows, because every early grab removes a row and a column from everyone else's options.

*Small grids are deceiving. The **model** is what scales, and it is the same three Solver cells with a grid of 0s and 1s in the changing cells.*

## In practice: real matching problems rarely arrive this tidy

---

### IN PRACTICE

**The greedy rule is wrong.** Grabbing the smallest entry can trap you into an expensive total, and handing each crew its own fastest run collides: Crews 1 and 3 both want the hub run. Let Solver match the whole grid at once.

**Make it square first.** More runs than crews? Add a *dummy crew* at zero cost so the model stays balanced; the run it takes is the one left undone. A dummy row records an unfilled run.

**The costs encode your real goal.** Hours in the grid minimize time; dollars minimize money; a preference penalty models which crew would rather not take a run. Make sure the numbers mean what you intend.

*Every one of the three models is bookkeeping for a real dispatch decision. The bookkeeping has to be honest before Solver is useful.*

## Three models, one Solver pattern

---

|             | Transportation                | Transshipment                        | Assignment                   |
|-------------|-------------------------------|--------------------------------------|------------------------------|
| Variables   | units on each route           | units on each arc, hubs included     | 0 or 1 on each pairing       |
| Constraints | row = supply, column = demand | the same, plus in = out at every hub | every row and column = 1     |
| Objective   | SUMPRODUCT of cost and flow   | the same, over every arc             | the same, over every pairing |
| Whole units | free, from the structure      | free                                 | free: clean 0s and 1s        |

*Learn one, and the other two are a change of right-hand sides and one extra equation.*

## The same grid after a flood

*Relief supplies from regional depots to shelters*

---

|                     | <b>The company</b>              | <b>After a flood</b>                     |
|---------------------|---------------------------------|------------------------------------------|
| Sources             | Plants A and B                  | regional depots                          |
| Destinations        | Stores 1 and 2                  | shelters                                 |
| $x_{ij}$            | units on a route                | cases of water, depot $i$ to shelter $j$ |
| $c_{ij}$            | \$ per unit                     | fuel, travel time, and road risk         |
| A route that is out | deleted, or shipment fixed at 0 | a closed bridge, handled the same way    |

---

*The same equalities account for every case shipped and every shelter's need.  
The grid is bookkeeping for a real incident command decision.*

## In practice: real shipping problems rarely arrive this tidy

---

### IN PRACTICE

**Unbalanced totals flip a constraint type.** When supply exceeds demand the supply rows become  $\leq$ : some stock stays put. When demand exceeds supply the demand columns become  $\leq$ : some demand goes unmet. Or add a *dummy* destination or source at \$0 cost; same plan either way, and its column reports which source holds stock back.

**Costs are not always per unit.** A carrier that charges per truck or gives volume discounts breaks the clean  $c_{ij}$ , and the plain model needs adjusting.

**Transportation hands you whole units; most models do not.** Where there is no such structure, never round: say `int` or `bin` and let Solver search.

*Your take-home's first grid does not balance once the units are converted.  
Fixing that is the modeling step.*

## Back to the company

---

- **Use both:** Plant B's 80 go straight to Store 1, and all 120 of Plant A's go through the hub, 20 on to Store 1 and 100 on to Store 2. That costs **\$1,020**, against \$1,160 direct only and \$1,100 through the hub only, and the hub takes in 120 and sends out 120.
- **Crew 1 on the Store 1 run and Crew 2 on the Store 2 run**,  $7 + 5 = 12$  hours. The cross assignment takes 17, and the slider's Least cost lands on 12.
- Same minimize-cost idea as Monday: a hub adds one equation, and a matching is the special case where every supply and every demand is 1.

*Two crews can be costed by hand. At four the greedy grab already costs 30 hours against 22, and at ten the hand method is not an option.*

## This week's take-home

*One file, on Canvas, by Sunday at 11:59 PM*

---

- **Network 1:** a book distributor whose depots count pallets while its zones order cases. Convert, discover that supply and demand **do not balance**, size the dummy destination, and let Solver fill the grid.
- **Network 2:** three library auditors to three branches. Try the greedy shortcut first and record what collides, then build the 0/1 grid properly.
- **Network 3:** a food-bank cross-dock. Write the **flow-balance cell** yourself, then solve the network three ways: direct only, dock only, both open.

ON THE SITE · the brief and the workbook

Download the workbook here. **Tier 1:** AI as a tool, disclosed. Solver fills the grids; the conversions, formulas and the repair are yours.

[v348.jima.me/week/w10/](https://v348.jima.me/week/w10/)



## Carry into Week 11

---

**\$1,020**

the hub and the direct  
routes together

**in = out**

one equation per hub

**22, not 30**

the greedy grab costs  
eight hours more

**10!**

why the model, not the  
eye, scales

*Next week a **legal-aid nonprofit** weighs cost, access and space for its site, then balances service goals: scoring models and goal programming with priorities.*

# Review

Week 10

## Can you define these without looking?

---

|                             |                     |                    |                      |
|-----------------------------|---------------------|--------------------|----------------------|
| transportation model        | source, destination | unit shipping cost | balanced, unbalanced |
| dummy source or destination | degree of freedom   | transshipment      | flow balance         |
| assignment model            | binary variable     | the greedy rule    | integer programming  |
| LP relaxation               | total, 0–1, mixed   | branch and bound   | int and bin          |

ON THE SITE · Key Concepts

Most of these have a one-line definition and a worked example behind them. Filter by week; the next two frames define most of them.

[v348.jima.me/concepts/](http://v348.jima.me/concepts/)



## Definitions: the three networks

---

|                             |                                                                                                          |
|-----------------------------|----------------------------------------------------------------------------------------------------------|
| <b>Transportation model</b> | Sources with supplies, destinations with demands, a unit cost per route. Minimize the cost sum           |
| <b>Balanced, unbalanced</b> | Supply equals demand, so both are equalities. If not, the larger side becomes $\leq$ , or a dummy at \$0 |
| <b>Degree of freedom</b>    | A balanced 2 by 2 has one free number; the rest is forced                                                |
| <b>Transshipment</b>        | A transportation model with hubs: source to hub, hub to hub, or direct                                   |
| <b>Flow balance</b>         | At a hub, inflow equals outflow: one Solver constraint per hub                                           |
| <b>Assignment model</b>     | Every supply and demand is 1 and every variable is 0 or 1: a matching                                    |
| <b>The greedy rule</b>      | Grab the cheapest cell. Not a method: 30 hours, not 22                                                   |

---

## Definitions: whole units

---

|                             |                                                                                                                                              |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Whole units for free</b> | Grids with whole supplies and demands return whole shipments, and clean 0s and 1s, by their structure alone                                  |
| <b>Integer programming</b>  | A linear program whose variables must be whole, because half a van does not exist                                                            |
| <b>LP relaxation</b>        | The model solved without whole-number restrictions: a ceiling for maximization, a floor for minimization                                     |
| <b>Rounding</b>             | Not a method. Up can break a limit; down can be legal and still not the best                                                                 |
| <b>Total, 0–1, mixed</b>    | Every variable whole; every variable a binary yes or no; some whole, some continuous                                                         |
| <b>Branch and bound</b>     | How Solver searches once <code>int</code> or <code>bin</code> is set: relaxed programs, discarding branches that cannot beat the best so far |

## Test yourself

*If one of these is slow, that is where to reread*

---

1. Plant A ships 120, Plant B 80, and the stores want 100 each. Why can you choose only one shipment freely, and what stops it going below 20?
2. After converting cases to pallets, supply exceeds demand. What do you add to the model, and at what cost?
3. With the hub open, Plant B sends Store 1 its 80 direct while Plant A sends all 120 through the hub. Why, and what single equation must the hub satisfy?
4. Why does an assignment grid return 0s and 1s without `int`, while the truck-and-van model needs it?
5. The relaxation says 3 box trucks and 1.5 cargo vans. Why is  $(3, 2)$  illegal,  $(3, 1)$  legal but not best, and  $(4, 0)$  the answer?

# Numbers worth remembering

---

**\$1,160**

two plants, two stores;  
20 on the free route

**\$1,300**

three plants, three  
stores, from Solver

**\$1,020**

the hub and the direct  
routes together

**22, not 30**

four crews: the  
optimum against the  
greedy grab

---

## Question

## This week

## The move

Cheapest plan on a 2 by 2

\$1,160

cost is  $6t + 1,040$ ; take the smallest  
legal  $t$ , 20

What a hub adds

one equation

inbound minus outbound, set to 0

Best whole-number plan

(4, 0)

not a rounding: search the lattice, or  
say `int`

---

## Carry into Week 11

---

- Week 11 follows a **legal-aid nonprofit**: cost, access and space compete when it chooses a site, and its service goals compete for staff time.
- **Goal programming** keeps the linear program and turns each goal into a target with a deviation to minimize, in order of priority or by weight.
- **Scoring models** do the same for a site or a vendor with no linear program at all: criteria, weights, and a ranking that flips when the weights move.

ON THE SITE · Week 11

The reading and the before-class self-check are posted. Do the self-check before Monday.

[v348.jima.me/week/w11/](https://v348.jima.me/week/w11/)

