Formulation

- Essential step in modeling
  - Abstracts the operational problem into a mathematical model
  - Is the first opportunity to test model validity
- **In optimization, the formulation is where the ambiguity ends**
- So how do you learn to formulate?
  - Practice, practice, practice
  - Formulation is also an art; real-life problems always have alternative formulations

Constructive Formulation Approach

- From Schrage (1997)
  - Determine what is to be decided (variables)
  - Determine how the decisions will be scored (objective function)
  - Determine conditions and relationships that restrict values of variables (constraints)
  - Populate model with data, or adjust for availability of data
  - Choose solution method appropriate for relationships
    - If relationships are too hard mathematically, consider adjusting model to give up precision for tractability
- Objective: train you to be able to employ this approach

Template Formulation Approach

- From Schrage (1997); is also Winston's approach
  - Start with a taxonomy of model types
  - Classify your situation according to this taxonomy
  - Use an existing model as a template for your problem
- Templates we will cover (for LP)
  - Product mix
  - Covering, staffing, scheduling
  - Blending
  - Multiperiod planning
  - Simple recourse (stochastic) models

  - Network models
  - Project planning models
  - (some) Two-sided game models

How You Will Formulate: NPS Format

- Accurate documentation is crucial
  - Lack of it has killed many projects
  - Subject treated poorly or omitted in mainstream texts (including Winston)
- The following format was popularized at the Naval Postgraduate School
  - Matches up very well with algebraic languages such as MPL
  - Acceptable to any journal
- Warning: the format is algebraic!
  - The max $c_1x_1+c_2x_2+ c_3x_3$ jazz is NOT allowed
  - Will force you to write compact, flexible formulations
  - Will make transition to large models painless

The Format

- Indices
  - Domains and fundamental dimensions for the model
  - Example: products, time periods, regions, factories
- Data
  - The input to the model, indexed using the indices
  - Convention: data is UPPERCASE
- Variables
  - The quantities to be determined, indexed using the indices
  - Convention: variables are lowercase

The Format (cont'd)

- Objective function
  - The quantity to be optimized
  - Indicate max or min; designate a variable = to the objective
- Constraints
  - The binding relationships
  - Constraints are ALSO indexed (real power of algebraic language)
  - Attach a word description to each set of constraints
  - Include bounds on variables (like nonnegativity)
- **ALGEBRAIC MODELING LANGUAGE CODE IS NOT A FORMULATION!**

Product Mix Using NPS Format

- Go back to Wyndor Glass
- Indices
  - ***p*** = products {1,2}
  - ***f*** = factories {1,2,3}
- Data
  - $\boldsymbol{PROFIT_p}$ = $ profit per unit of *p* sold
  - $\boldsymbol{CAP_{pf}}$ = capacity required per unit of *p* built at *f*
  - $\boldsymbol{TOTCAP_f}$ = total capacity available at *f*
- Variables
  - $\boldsymbol{num_p}$ = units of *p* to produce
  - ***totprofit*** = total profit

Product Mix NPS Format (cont'd)

- Objective
  - $\max_{num} \quad totprofit = \sum_{p} PROFIT_p * num_p$
- Constraints
  - $\sum_{p} CAP_{pf} * num_p \leq TOT_f$ for all f (factory capacity constraints)

    $num_p \geq 0$ for all p (nonnegativity)

- This is *compact, scalable*, and *easily implementable*
  - Works for **2** products and **3** factories, or **m** products and **n** factories
  - Uses index, variable, and data names that relate to the problem

Characteristics of the Product Mix Problem

- Set of "products" that could be produced
- Products require differing amounts of limited resources
- Products have different costs, profits, or demands
- Problem is generally static - no time dimension
- Challenge for the students
  - Suppose in the Wyndor Glass problem, sales beyond the first $INITIAL_p$ units have a $DISCOUNT_p$ profit decrease due to discounting
  - How do we account for that in the model?

Formulation II: Covering, Staffing, Scheduling

- Covering problems
  - Some set of activities have to be “covered”
  - Normally looking for a minimum cost solution
- Staffing and scheduling are essentially covering problems
- Can take different forms
  - Do the best with what you have to work with (optimize performance)
  - Determine needed resources (optimize design)
- Warning
  - Most real problems require integral answers; LP doesn’t work well
  - Many scheduling problems are real backbreakers
  - Be very careful when taking on a big covering problem

Example: Winston p. 75, #3 and #4

- Read problem description - any ambiguities?
  - Does an employee always work overtime?
  - Do we have to know how much of the requirement/day is regular and how much is overtime?
  - Does the day of overtime always occur at the end of the regular 5-day shift? Before? Either? Does it matter?
- Data as presented
  - $50/day for straight time, $62/day for overtime
  - Daily requirements
    - Monday - 17; Tuesday - 13
    - Wednesday -15; Thursday - 19
    - Friday - 14; Saturday - 16
    - Sunday -11

Get Something Down on Paper

- Indicies
  - **d** = days {m,t,w,th,f,s,sn}
- Data
  - $\mathbf{REQ_d}$ = workers required per day
  - **SCOST** = $ per week per worker for straight time ($250)
  - **OCOST** = $ per week per worker for overtime ($312)
- Variables?
  - $\mathbf{s_d}$ = number of workers starting on day **d** working straight time
  - $\mathbf{o_d}$ = number of workers starting on day **d** working overtime
  - **totcost** = total weekly cost to be minimized
  - Will this work? What else will we need to do?

Objective and Constraints

- Objective
  - $$\min_{s,o} \quad totcost = SCOST * \sum_d s_d + OCOST * \sum_d o_d$$
- OK, smart guy, how do you write *these* constraints algebraically?
  - Answer #1: attach a number to each day, then come up with some function that maps day starting to days covered
  - Answer #2: define a multidimensional *set,* and sum over that
- We'll go with #2
  - Define **scover(d,d1)** as all the days d1 covered by a straight-time worker starting on day d
  - Define **ocover(d,d1)** the same way
  - **d1** is called an *alias* for **d**; they both index the same set

Continuing ...

- So, the **scover(d,d1)** set would look like:
  - {m,m},{m,t},{m,w},{m,th},{m,f}
    {t,t},{t,w},{t,th},{t,f},{t,s} …
  - ocover(d,d1) is similar
  - NOTE: it's much easier to define sets of days NOT covered; also, we could use (d-1) for overtime shifts if we define d as "circular"
- The constraints (note **d1** and **d**!):

$$\sum_{d \in scover(d,d1)} s_d + \sum_{d \in ocover(d,d1)} o_d \geq REQ_{d1} \quad \text{for all } d1$$

$$s_d, o_d \geq 0 \quad \text{for all } d$$

Constraints in Variable-By-Variable Form

$$s_m + s_{th} + s_f + s_s + s_{sn} +$$
$$o_m + o_w + o_{th} + o_f + o_s + o_{sn} \geq REQ_m \quad \text{(Monday)}$$

$$s_m + s_t + s_f + s_s + s_{sn} +$$
$$o_m + o_t + o_{th} + o_f + o_s + o_{sn} \geq REQ_t \quad \text{(Tuesday)}$$

$$\vdots$$

$$etc$$

So What's the Answer?

- Employees
  - 5-day shifts: 2 Tuesday, 4 Thursday, 3 Sunday
  - 6-day shifts: 6 Monday, 2 Wednesday, 2 Saturday
  - Note: LP solution was naturally integer!
- Total cost: 5370; cheaper than original solution?
  - How much overtime pay is going out?
  - How would you modify this to limit overtime pay?

How about #4?

- Use same indices and data (assume no overtime)
- Variables?
  - s_d = number of workers starting on day ***d*** working straight time
  - ***totdays*** = total weekend days off (to be maximized)
- Objective
  - $\max_{s} \quad totdays = 2 * s_m + s_t + s_{sn}$
- Constraints

$$\sum_{d1 \in scover(d1,d)} s_{d1} \geq REQ_d \text{ for all } d$$

$$\sum_{d} s_d = 25$$

$$s_d \geq 0 \text{ for all } d$$

The Answer Is...

- 23 total weekend days
- Shift assignments
  - Monday: 6
  - Tuesday: 8
  - Thursday: 2
  - Friday: 6
  - Sunday: 3
- LP produced natural integer answer again!

Formulation III: Blending Problems

- Were the earliest problems attacked with LP
  - Stigler’s diet problem (1945) predated Dantzig’s simplex work
  - Heavily used by oil companies, agricultural firms
- Characteristics
  - Problem starts with a set of input raw materials
  - Each raw material has some set of qualities
  - Materials must be blended so the outputs have certain aggregate qualities
  - In linear form, assumes that output quality is some weighted average of the input quality

Winston p. 94, #14

- Indicies
  - **g** = gasolines {r,p}
  - **i** = inputs {ref, fcg, iso, pos, mtb, but}
- Data
  - $\mathbf{AVAIL_i}$ = daily availability of input i in liters
  - $\mathbf{RON_i}$ = octane of input i
  - $\mathbf{RVP_i}$ = RVP rating of input i
  - $\mathbf{A70_i}$ = ASTM volatility of i at 70C
  - $\mathbf{A130_i}$ = ASTM volatility of i at 130C
  - $\mathbf{RONRQ_g}$ = required octane of gas g
  - $\mathbf{RVPRQ_g}$ = required RVP rating of gas g
  - $\mathbf{A70RQ_g}$ = ASTM volatility of g at 70C required
  - $\mathbf{A130RQ_g}$ = ASTM volatility of g at 130C required

Blending: p. 94, #14 (cont'd)

- Data (cont'd)
  - **$DEMAND_g$** = daily minimum demand for gas g
  - **$PRICE_g$** = selling price/liter of gas g
  - **FCGLIM** = limit on proportion of FCG in each gas g
  - Do we need to include the lead removal cost in the LP? Again, what are we trying to decide?
- Variables
  - ***inp_{gi}*** = liters of input i used to make gas g
  - ***totgross*** = total gross from gas sales
- Objective function
  - $$\max \quad totgross = \sum_{g,i} PRICE_g * inp_{gi}$$

Blending: p. 94, #14 (cont'd)

- Constraints (easy)

$$\sum_{g} inp_{gi} \leq AVAIL_i \text{ for all } i \text{ (don't exceed availability)}$$

$$\sum_{i} inp_{gi} \geq DEMAND_g \text{ for all } g \text{ (meet demand)}$$

- Harder constraints

$$\frac{inp_{g,\text{"}fcg\text{"}}}{\sum_{i} inp_{gi}} \leq FGCLIM \text{ for all } g \text{ (proportional limit)}$$

linearize :

$$inp_{g,\text{"}fcg\text{"}} \leq FGCLIM * \sum_{i} inp_{gi} \text{ for all } g$$

Blending: p. 94, #14 (cont'd)

- Hardest constraints

$$\frac{\sum_i RON_i * inp_{gi}}{\sum_i inp_{gi}} \geq RONRQ_g \quad \text{for all } g \text{ (meet octane limit)}$$

$$\text{linearize:}$$

$$\sum_i RON_i * inp_{gi} \geq RONRQ_g * \sum_i inp_{gi} \quad \text{for all } g$$

- Remainder left as an exercise (but what about RVP? Is it a min, equality, or what?)

The Rest of the Constraints

$$\sum_i RVP_i * inp_{gi} = RVPRQ_g * \sum_i inp_{gi} \text{ for all } g$$

$$\sum_i A70_i * inp_{gi} \geq A70RQ_g * \sum_i inp_{gi} \text{ for all } g$$

$$\sum_i A130_i * inp_{gi} \geq A130RQ_g * \sum_i inp_{gi} \text{ for all } g$$

Formulation III: Multiperiod Planning

- Modeling partitioned by time periods
  - Some decision to be made in each time period
  - Decisions cover some time horizon
- Typical examples
  - Inventory models
  - Financial models, such as cash flows
  - Multiperiod work scheduling
- Formulation challenges
  - Determining linkages between time periods
  - Deciding whether to discount across time
  - Handling “end effects”

Inventory example: Winston p. 104, #5

- Indices
  - **t** = time {1,2} (NOTE: t is *always* time, if your model uses time)
  - **v** = vehicle type {car, truck}
- Data
  - $\mathbf{DEMAND_{vt}}$ = demand for v in month t
  - $\mathbf{LIMIT_{t}}$ = maximum vehicle production in month t
  - $\mathbf{STEEL_{v}}$ = tons of steel required for vehicle v
  - $\mathbf{SCOST_{t}}$ = cost per ton of steel in month t, $
  - $\mathbf{SAVAIL_{t}}$ = tons of steel available in month t
  - $\mathbf{BINV_{v}}$ = beginning inventory of vehicle v
  - **HOLD** = holding cost per vehicle per month, $
  - $\mathbf{MPG_{v}}$ = miles per gallon of vehicle v
  - **MPGAVG** = required avg MPG for all vehicles produced each month

Inventory, continued

- Variables: what has to be decided?
  - **$prod_{vt}$** = number of v produced in month t
  - **totcost** = total cost of meeting demand (holding plus steel)
  - Do we need anything else?
  - **inv_{vt}** = inventory of v at the end of month t
  - NOTE: the inventory variables are a convenience; we could formulate the problem without them (and in integer programming applications, that might be better). We will use them for clarity
- Objective
  - minimize cost of holding plus cost of steel
  -

$$\min_{prod,inv} \quad totcost = \sum_{v,t} SCOST_t * STEEL_v * prod_{vt} + HOLD * \sum_{v,t} inv_{vt}$$

Inventory, cont'd

- Easy constraints

$$\sum_{v} prod_{vt} \leq LIMIT \quad \text{for all } t \text{ (production limits)}$$

$$\sum_{\mathrm{v}} STEEL_{v} * prod_{vt} \leq SAVAIL \quad \text{for all } t \text{ (steel purchase limits)}$$

- Harder constraints

$$\frac{\sum_{v} MPG_{v} * prod_{vt}}{\sum_{v} prod_{vt}} \geq MPGAVG \quad \text{for all } t \text{ (MPG fleet limit)}$$

linearize:

$$\sum_{v} MPG_{v} * prod_{vt} \geq MPGAVG * \sum_{v} prod_{vt}$$

Inventory, cont'd

- The hardest part: material balance constraints
  - In words: inventory from last period + production - demand = end of period inventory
- So:

$$BINV_v + prod_{vt} - DEMAND_{vt} = inv_{vt} \text{ for all } v,t = 1$$

$$inv_{v,t-1} + prod_{vt} - DEMAND_{vt} = inv_{vt} \text{ for all } v,t > 1$$

$$inv_{vt}, prod_{vt} \geq 0 \text{ for all } v,t$$

- Does this guarantee demand will be met? How?
- Suppose steel could be held across periods? How do we handle that?

There Is One Central Trick in These Models

- These formulations typically use extra variables
  - Represent some resource carried from one period to the next
  - Are a function of activity in previous period and current period
  - Makes formulation clearer (and less dense)
- How would we substitute out the **inv$_{vt}$** variables?

$$BINV_v + prod_{vt} - DEMAND_{vt} \geq 0 \text{ for all } v,t = 1$$

$$\left(BINV_v + prod_{vt-1} - DEMAND_{vt-1}\right) + prod_{vt} - DEMAND_{vt} \geq 0 \text{ for all } v,t = 2$$

$$\left[\left(BINV_v + prod_{v,t-2} - DEMAND_{v,t-2}\right) + prod_{v,t-1} - DEMAND_{v,t-1}\right] + prod_{vt} - DEMAND_{vt} \geq 0 \text{ for all } v,t = 3$$

$$\vdots$$

$$BINV + \sum_{t1 \leq t}\left(prod_{v,t1} - DEMAND_{v,t1}\right) \geq 0 \text{ for all } v,t$$

Model Effects By Substituting Out

- Suppose we have **N** inventory balance constraints
- With explicit inventory variables:
  - **N** production + **N** inventory = **2N** variables
  - Also have **2N** nonzero coefficients in the constraints
- If you substitute them out:
  - **N** production =**N** variables
  - However, have **N** + **(N-1)** + **(N-2)** + … + 1 = **N(N+1)/2** nonzeros
- At N=20:
  - 40 variables and 40 nonzeros with explicit inventory variables
  - 20 variables and 210 nonzeros by substituting out
- Former is better for LP, latter is better for IP if production variables are integer (I will say why this is so later)

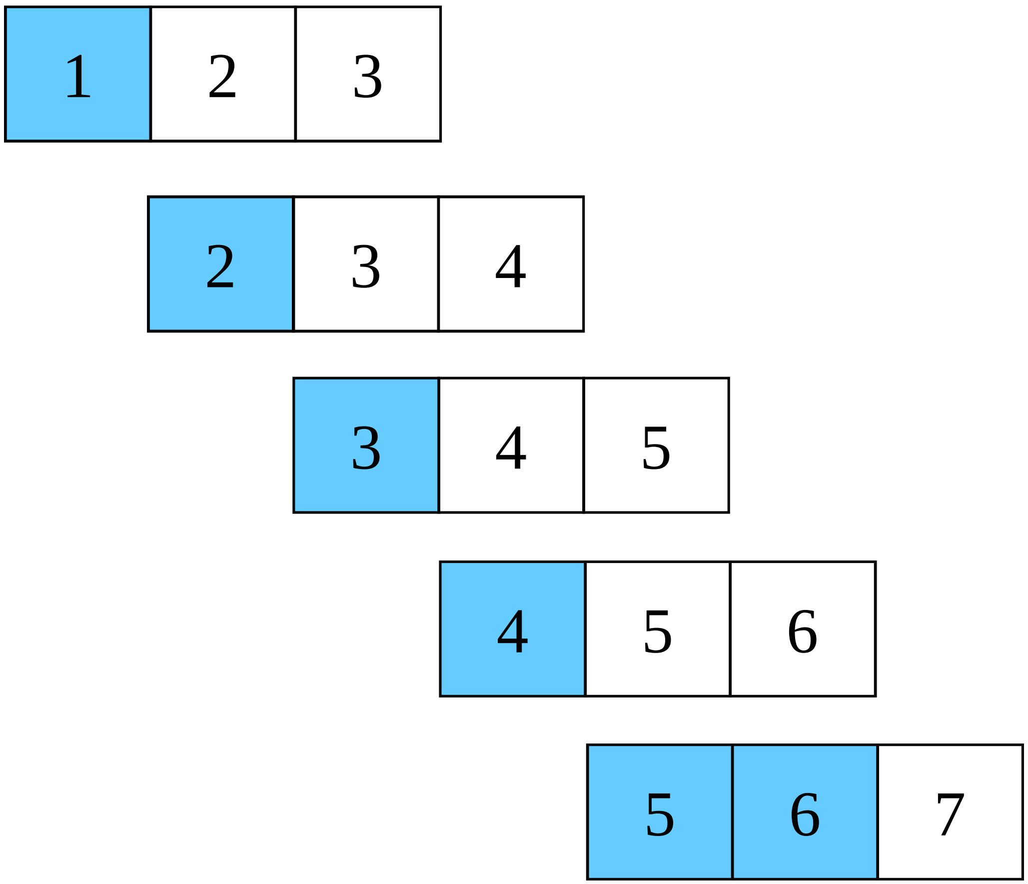
Modeling Issues with Multiperiod Models

- Demand certainty
  - Future demands, prices, costs, etc are almost always random
  - Yet, an LP must treat them as certain
  - Seems unreasonable not to account for this
- Model Omniscience
  - LPs pursue extreme solutions
  - In a multiperiod model, you are giving the optimization perfect knowledge of the future
  - Can lead to very strange behaviors (end effects; relate DAWMS story)

Some Tricks to Address These Issues

- Discounting
  - Idea here is to "discount" impact of decisions made in future periods
  - In design problems, gives more weight to more certain demands, prices, conditions
  - How would discounting apply to the car example?
- Cascading
  - An excellent technique, not used enough
  - Handles cases where the model has to allocate resources *across* time, but behaves badly if it knows the future
  - Method: break model into a sequence of LPs that "cascade" across time
  - Run model for **n** periods, "freeze" results for some **m < n** periods, and run the next **n**-period solution

Cascade Example



- Convert a 6-period model into a 3-period model, with 5 separate runs
- In each run but the last, the results of the first period are fixed, and resources used there are subtracted in the next run
- The model always sees the future, but its horizon is limited
- It also thinks there's some future after period 6