15.053 Exam 2 Notes

Linear Programming – The End

- Sensitivity analysis
  - *Relaxing or removing/tightening or adding* constraints makes *more/less* solutions available. This can be done by modifying the LHS or RHS.
  - Similarly *adding/removing* variables makes *more/less* solutions available.
  - *Less/more* feasible solutions means that the optimal value must stay the same or get *worse/better*.
  - It turns out, however, that this effect is **not** linear.
  - Changing an objective function coefficient will also change the result. It's also non-linear, but that's less obvious.
- LP Duality
  - Every LP has a dual, which characterises the sensitivity of the original solution.
  - One dual variable for each main LP constraint – **change in primal optimal value per unit increase in constraint RHS**.
  - In a way, it's the "fair price" of unit thing on the RHS of the constraint. We also call it the **marginal price** or **shadow price**.
  - Marginal/shadow price can be worked out from a simple linear program by increasing the RHS of a constraint by Δ, manually finding the new optimal solution, and seeing by how much in improves the objective function.
  - Marginal/shadow price can be worked out from a Simplex tableau. Simply work out the reduced cost for taking a slack variable into the basis, multiply it by –1, and this gives the shadow price.

- Can also work out from Simplex tableau what range certain objective function coefficients can take before the current optimal solution stops being optimal.
  - Find every improving direction
  - Require that the reduced cost for every improving direction be **bad** – otherwise, the current optimal solution is no longer optimal, because we could go along the "good" directions.

- Constructing the dual
  - Consider an LP with n variables, m constraints:

$$\min \sum_{j=1}^{n} c_j x_j$$
$$\text{s.t.} \sum_{j=1}^{n} a_{i,j} x_j \geq = \leq b_i$$
$$x_j \geq = \leq 0$$

  - The dual will have
    - One decision variable v_i for each constraint
    - Objective function $\max \sum_{i=1}^{m} b_i v_i$
    - For each **primal decision variable**, a constraint of the form

$$\sum_{i=1}^{m} a_{i,j} v_i \leq = \geq c_j$$

      [Going "down the columns" of the primal problem]. The type of each constraint depends on the non-negativity constraint of the original variable.
    - For each **dual decision variable**, make it $\geq$/$\leq$/free if the original inequality was $\geq$/$\leq$/$=$
  - Swapping signs
    - **MINIMISATION PROBLEM**: keep signs of inequality going from *constraint* → *non-negativity*, and flip the sign when going from *non-negativity* → *constraint*.
    - **MAXIMISATION PROBLEM**: vice-versa.

- **Strong duality** – the optimal value of the primal and dual are the same.

- **Primal complimentary slackness** – *either* the optimal solution makes a main inequality constraint active *or* the corresponding dual variable has optimal value = 0. This reflects the fact that if something is not in short supply in the primal, its shadow cost is 0.
- **Computer output**
  - *Constraint sensitivity analysis* gives us
    - **Type** → $L \Leftrightarrow \geq$, $U \Leftrightarrow \leq$, $LU \Leftrightarrow =$
    - **Optimal dual** → optimal value of the dual variable corresponding to the constraint [= marginal]
    - **RHS coef** → the specified RHS coefficient in the original primal problem.
    - **Max increase/decrease** → the amount by which the RHS can be increased/decreased for which the dual solution remains valid.
  - *Variable sensitivity analysis*
    - **Optimal value** → optimal value of primal variable
    - **Bas Sts** → basic (B) or non-basic (NL/NU)
    - **Lower bound/upper bound/object coef** → specified stuff in the primal problem
    - **Reduced object** → |reduced cost of variable| at optimality. This is how much more attractive the variable's coefficient in the objective function must be before the variable is worth using.
    - **Max decrease/increase** → change of objective function coefficient for which primal optimal solution remains unchanged.

Games

- **0-sum game**: whatever someone wins, the other loses.
- **Pure strategy:** choose same thing every time. **Mixed strategy**: assign *probability* to each strategy.

- Want to **MAXIMISE** *your* **MINIMUM** payoff or want to **MINIMISE** *their* **MAXIMUM** payoff.
- Turns out these two aims/programs are **duals** of each other.

Graphs

- **Principle of optimality (for shortest path problem)** – in a graph with no negative dicycles (directed paths that start and end at the same node), optimal paths must have optimal subpaths.
- **Bellman-Ford algorithm**
  - Set $v = 0$ for the source, ∞ otherwise.
  - At each iteration, set v as the **minimum** of
    - Its previous value
    - For any node that leads into it, the v for that node + the length of the path that leads there.

    Set d to the neighbouring node that was used to work out v.
  - Terminate if
    - All the vs stay the same for one cycle
    - $t = n$ and they're not, which means there's a negative dicycle.
- The **critical path** is the **longest path through a graph**.