Day 11 Notes

Zach Neveu

May 22, 2019

1 Agenda

- LP solving with simplex
- Introduction to LP
- Homework #3 due Friday
- Quiz #3 next Tuesday

2 LP

Options for LP soln:

- Single possible solution
- No feasible solution
- Unbounded feasible region
- Infinite optimal solutions along line segment

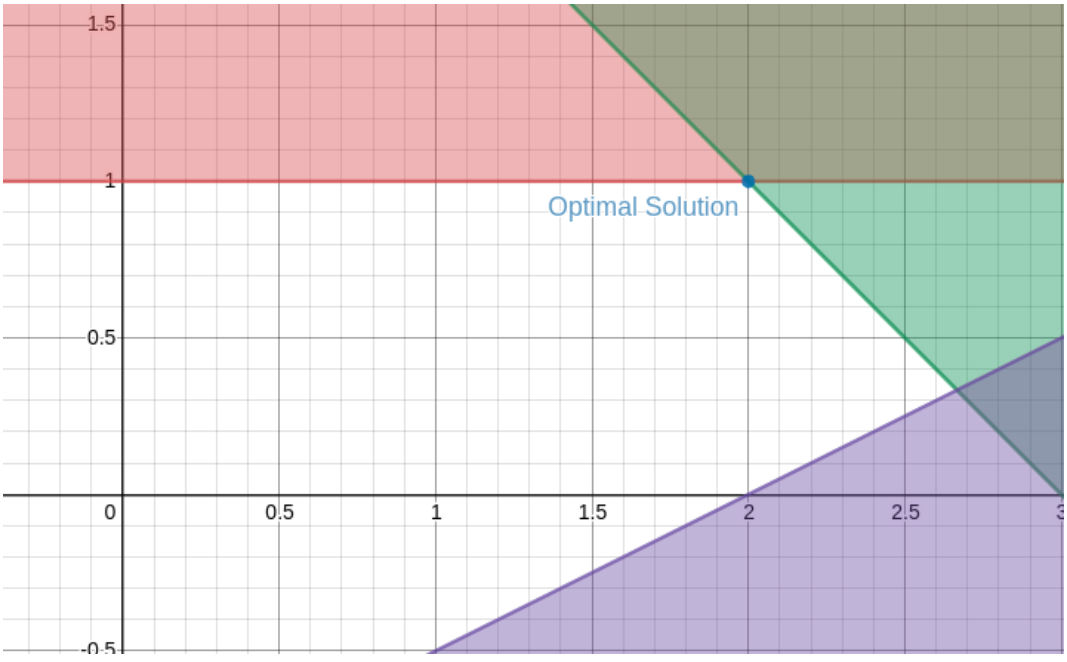

Figure 1: Single Solution

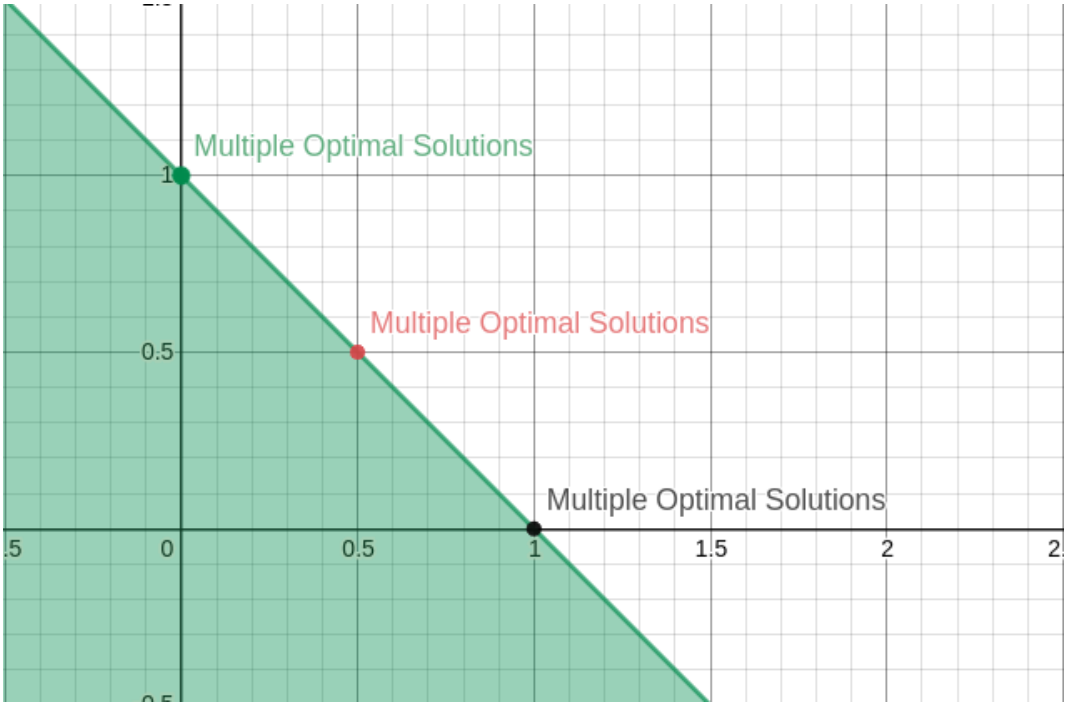

Figure 2: Multiple Solutions Along Segment

- **Cornerpoint Feasible Solution (CPF Solution):** Cornerpoint solutions that border the feasible region.
- Number of cornerpoints $\leq \binom{constraints}{2}$
- Adjacent CPF solutions are connected along the edge of a constraint
- Optimality Test: If a CPF Solution is better than all adjacent CPF solutions, then it must be optimal.
- This works because simplex is always **convex**
- **Convex:** Lines connecting all pairs of nodes fall completely within the enclosed area
- **Boundary** of the feasible region is the parts of the constraint boundaries that are feasible
- CPF solutions in 3 dimensions are the intersection of 3 planes
- In 3 dimensions, each CPF solution has up to 3 possible adjacent solutions
- **Edge:** The feasible portion of the intersection of 2 constraint boundaries
- Number of possible CPF solutions can be exponentially large (num constraints choose num dimensions)
- Not trivial to search through all possible CPF solutions in certain cases
- Bad news: Possible to create an intractable simplex instance
- Good news: In practice, instances are solved very fast
- More Good news: other algorithms can solve LP problems in P!

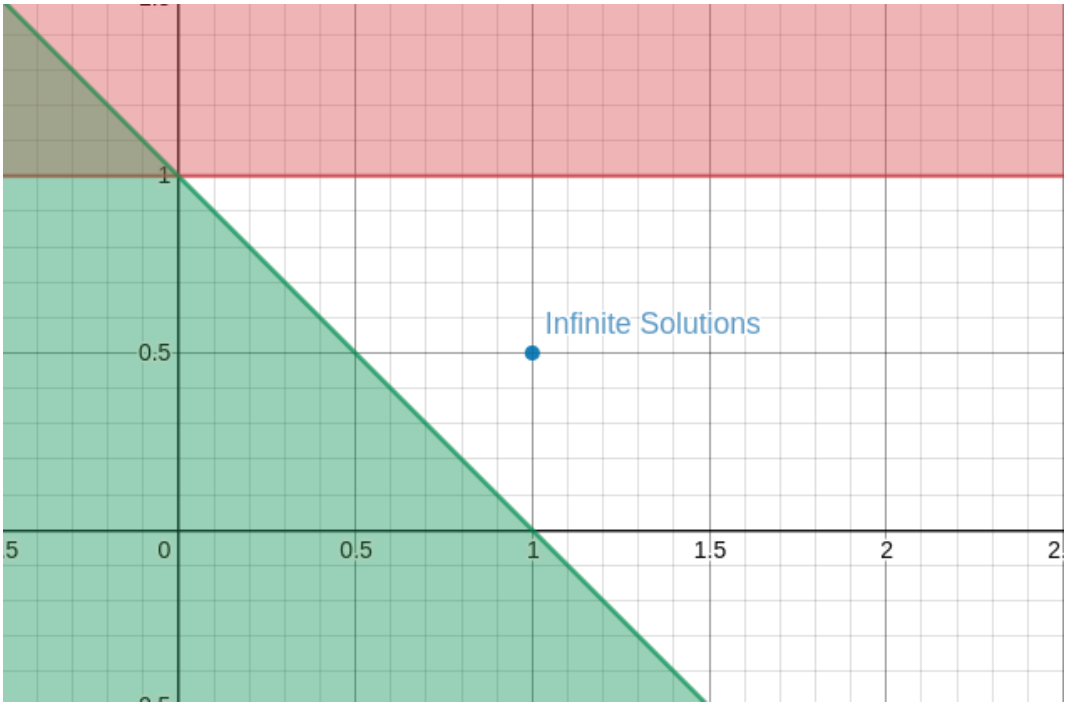

Figure 3: Unbounded Feasible Region

Simplex Algorithm

1. Initialization: pick an initial CPF Solution
2. Optimality test
3. Select new CPF Solution that is adjacent
4. Go back to 2 until optimality test passes.

3 Using LP in Class

AMPL Notes

```
# Write in *.mod file
var x1;
var x2;
maximize objective: x1+x2;
subject to constraint1:
    4*x1-x2 <= 18;
constraint2:
    2*x1+x2 <=  18;
```

1. Write AMPL commands into *.mod file
2. Run `ampl` at unix prompt to launch AMPL

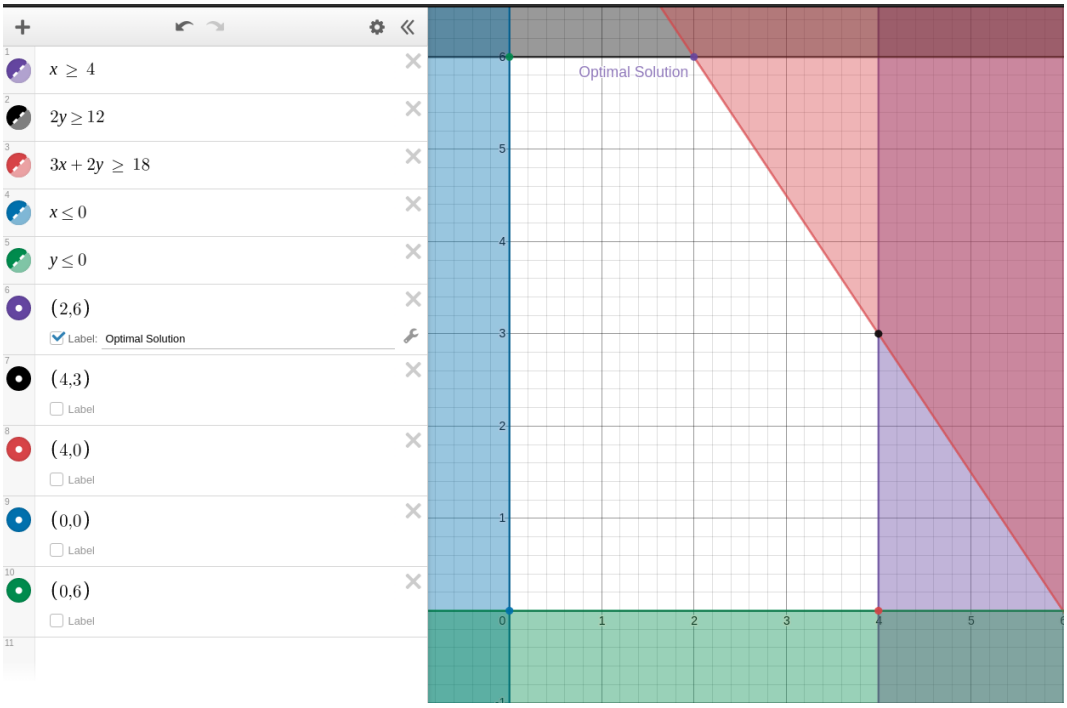

Figure 4: Example LP from Text Book

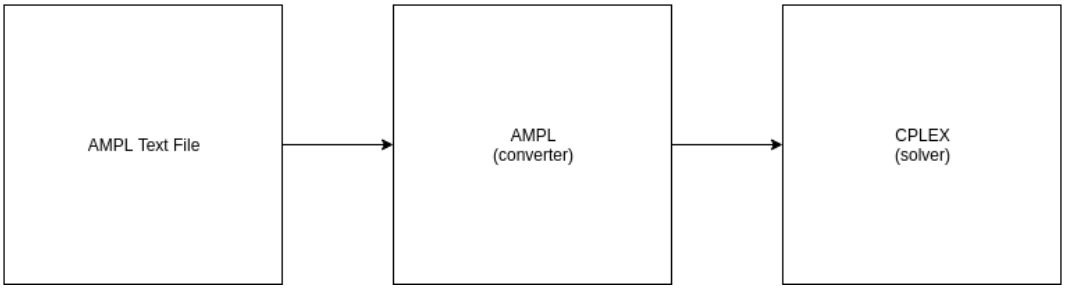

Figure 5: Flow Diagram of Solving Process

3. run `model *.mod` to load model
4. run `data *.dat` to load data
5. run `solve;` to solve
6. Solver spits out # of simplex steps required
7. run `display x1;` to display result
8. run `include *.run` to run script of AMPL commands
9. run `ampl *.run` from unix prompt to run script

AMPL Examples

```
param N=10;
var x {i in 0..N-1};
subject to C1:
    x[0]+3*x[i] <=  10;
c2:
    sum{i in 0..N-1} x[0] = 0;
c3 {j in 0..10}: x[j] = x[j+1];
```

- Variables ≠ Parameters
- Parameters are known at solve time - unchanging
- Variables: may change in execution
- Store params in data file
- Store variables in model file