Linear Programming

What is Linear Programming?

- A method for solving certain types of minimization/maximization problems.
- The problems must be expressible by a system of inequalities
- The inequalities must have very specific constraints.

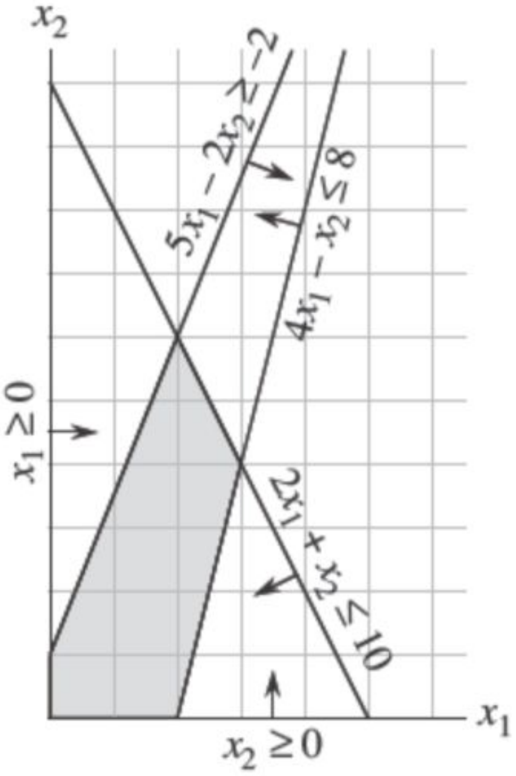
Our first example:

$$\begin{array}{ll} \text{maximize} & x_1 + x_2 \\ \text{subject to} & \end{array}$$

$$\begin{array}{rcr} 4x_1 - x_2 & \leq & 8 \\ 2x_1 + x_2 & \leq & 10 \\ 5x_1 - 2x_2 & \geq & -2 \\ x_1, x_2 & \geq & 0 \end{array}$$

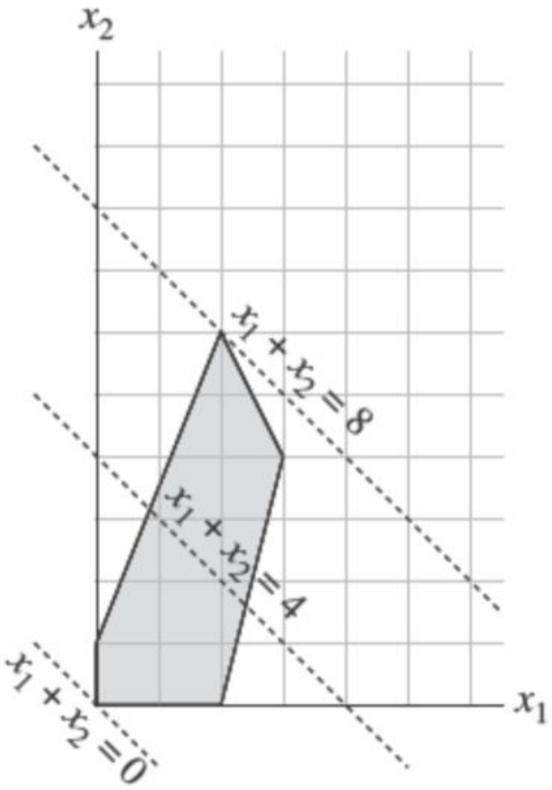
Introduction to Algorithms, 3rd ed
p. 846

Let's graph it!



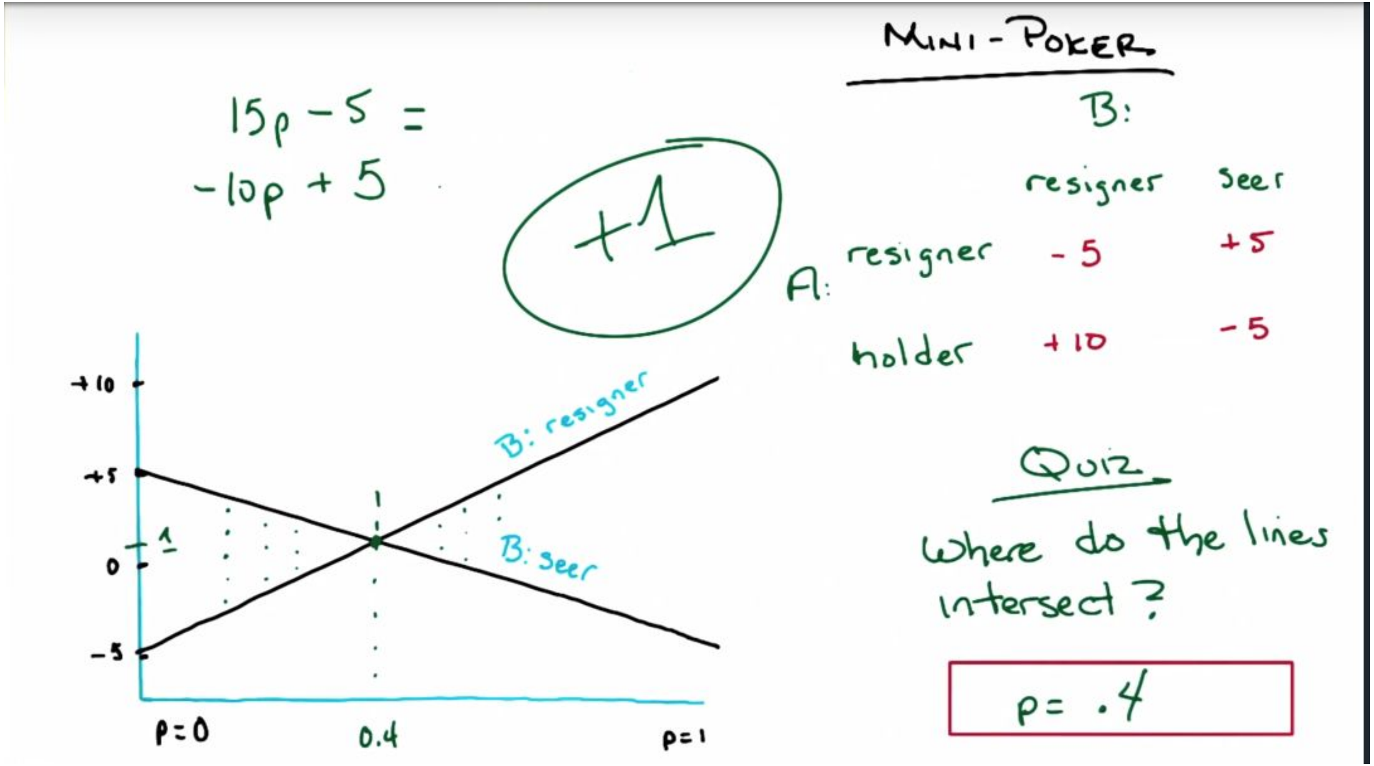
Introduction to Algorithms, 3rd ed
p. 847

And, we are done...



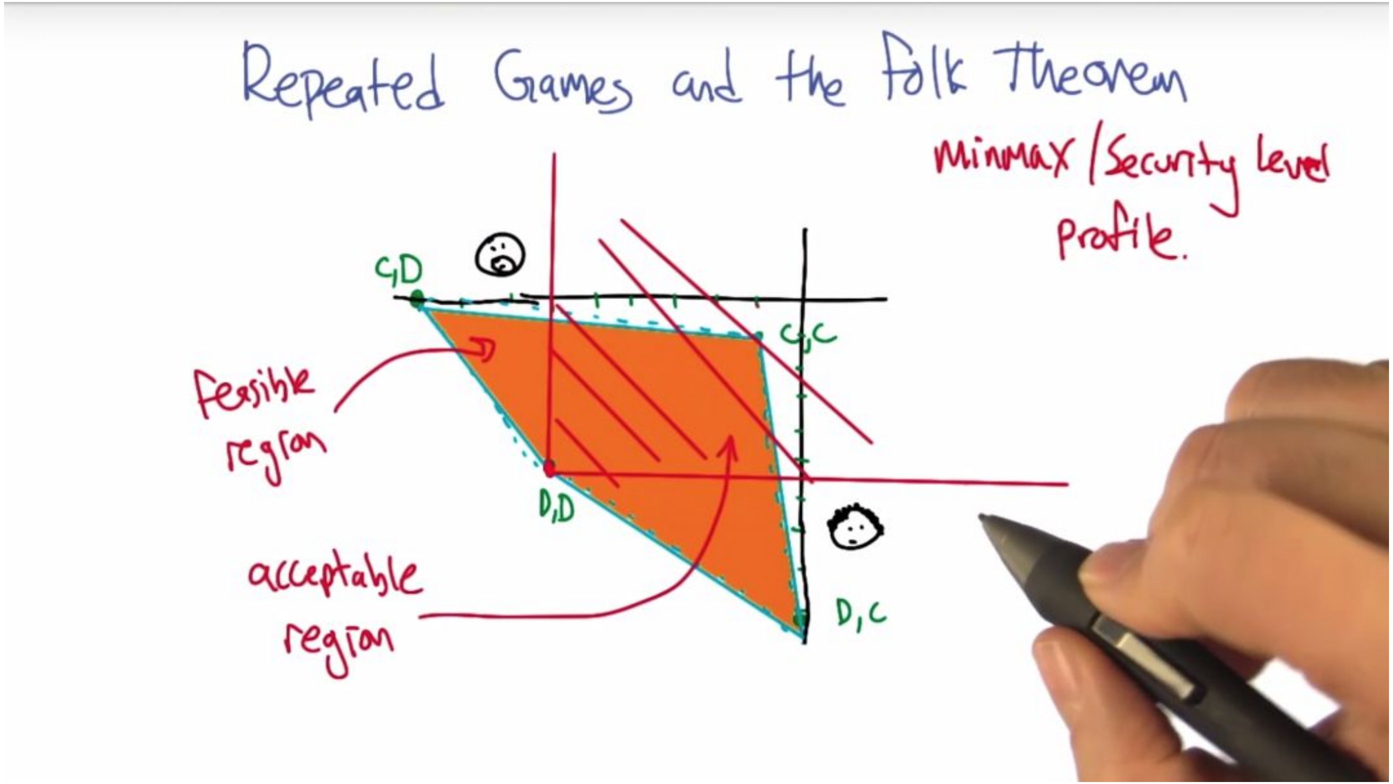
Introduction to Algorithms, 3rd ed
p. 847

This should look familiar



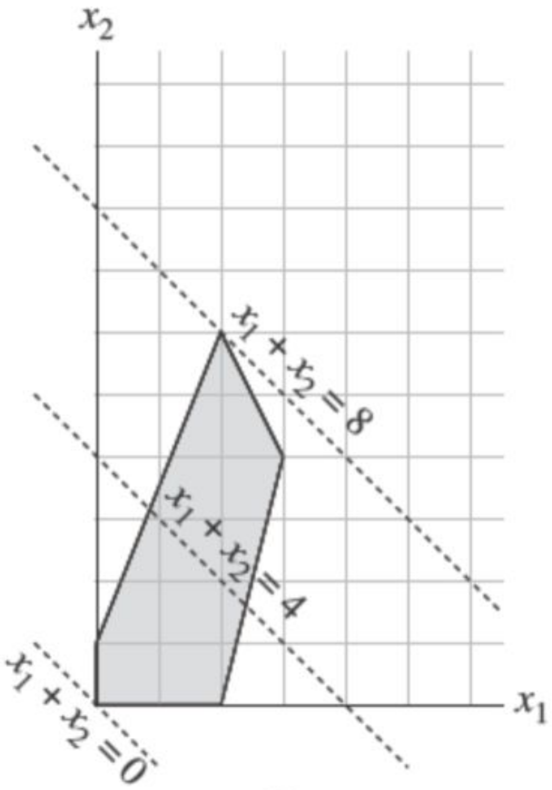
Udacity Reinforcement Learning
Lesson 12: 11A Game Theory
16. Center Game

Seriously, really familiar…



Udacity Reinforcement Learning
Lesson 13: 11B Game Theory Reloaded
13. Security Level Profile

So, how do we use a computer to solve this?



Introduction to Algorithms, 3rd ed
p. 847

Linear Programming!

- Get yourself a LP solver: glpk, CVXOPT, SCPSolver, etc...

- Express your problem as a system of inequalities.

- Run your solver on the resulting matrix.

Back to our example

$$\begin{array}{ll} \text{maximize} & x_1 + x_2 \\ \text{subject to} & \end{array}$$

$$\begin{aligned} 4x_1 - x_2 &\le 8 \\ 2x_1 + x_2 &\le 10 \\ 5x_1 - 2x_2 &\ge -2 \\ x_1, x_2 &\ge 0 \end{aligned}$$

Introduction to Algorithms, 3rd ed
p. 846

Almost there

$$
\begin{array}{ll}
\text{minimize} & -x_1 - x_2 \\
\text{subject to} & \\
& \begin{aligned}
4x_1 - x_2 &\leq 8 \\
2x_1 + x_2 &\leq 10 \\
5x_1 - 2x_2 &\geq -2 \\
x_1, x_2 &\geq 0
\end{aligned}
\end{array}
$$

Introduction to Algorithms, 3rd ed
p. 846

"Standard Form"

$$\begin{aligned} \text{minimize} \quad & -x_1 - x_2 \\ \text{subject to} \quad & \end{aligned}$$

$$\begin{aligned} 4x_1 - \quad x_2 &\leq \quad 8 \\ 2x_1 + \quad x_2 &\leq \quad 10 \\ -5x_1 + 2x_2 &\leq \quad 2 \\ x_1, x_2 \quad &\geq \quad 0 \end{aligned}$$

Introduction to Algorithms, 3rd ed
p. 846

CVXOPT LP Solver

http://cvxopt.org/userguide/coneprog.html#cvxopt.solvers.lp

cvxopt.solvers.lp(*c*, *G*, *h*[, *A*, *b*[, *solver*[, *primalstart*[, *dualstart*]]]])

Explicit Multipliers

$$
\begin{array}{ll}
\text{minimize} & -1x_1 - 1x_2 \\
\text{subject to} &
\end{array}
$$

$$
\begin{aligned}
4x_1 - \ 1x_2 &\leq \ \ 8 \\
2x_1 + 1x_2 &\leq 10 \\
-5x_1 + 2x_2 &\leq \ \ 2 \\
x_1, x_2 \quad &\geq \ \ 0
\end{aligned}
$$

Introduction to Algorithms, 3rd ed
p. 846

CVXOPT LP Parameters

$$\text{minimize} \quad \underbrace{-1x_1 - 1x_2}_{c}$$

$$\text{subject to}$$

$$\begin{array}{rrrcr}
\text{G} & & \text{G} & & \text{h} \\
4x_1 & - & 1x_2 & \leq & 8 \\
2x_1 & + & 1x_2 & \leq & 10 \\
-5x_1 & + & 2x_2 & \leq & 2 \\
 & x_1, & x_2 & \geq & 0
\end{array}$$

Introduction to Algorithms, 3rd ed
p. 846

CVXOPT LP Parameters

c $\boxed{[-1, -1]}$

G $\boxed{[4, 2, -5], [-1, 1, 2]}$

h $\boxed{[8, 10, 2]}$

cvxopt.solvers.lp(*c*, *G*, *h*[, *A*, *b*[, *solver*[, *primalstart*[, *dualstart*]]]])

Code

```
from cvxopt import matrix, solvers


c = matrix([-1., -1.])
G = matrix([[4., 2., -5.], [-1., 1., 2.]])
h = matrix([8., 10., 2.])

solution = solvers.lp(c, G, h)
```

https://github.com/axonal/cvxopt-tutorial/blob/master/clrs.py

Just a small bit harder… The game of "chicken"

	L	R
T	6,6	2,7
B	7,2	0,0

The correlated equilibria in this game are described by the probability constraints $\pi_{TL} + \pi_{TR} + \pi_{BL} + \pi_{BR} = 1$ and $\pi_{TL}, \pi_{TR}, \pi_{BL}, \pi_{BR} \geq 0$ together with the following rationality constraints:

$$-1\pi_{TL} + 2\pi_{TR} \geq 0 \qquad -1\pi_{TL} + 2\pi_{BL} \geq 0$$
$$1\pi_{BL} - 2\pi_{BR} \geq 0 \qquad 1\pi_{TR} - 2\pi_{BR} \geq 0$$

Correlated-Q Learning
Greenwald/Hall 2003

Four Inequalities

	3	4
	L	R
1 T	6,6	2,7
2 B	7,2	0,0

The correlated equilibria in this game are described by the probability constraints $\pi_{TL}+\pi_{TR}+\pi_{BL}+\pi_{BR}=1$ and $\pi_{TL},\pi_{TR},\pi_{BL},\pi_{BR}\geq 0$ together with the following rationality constraints:

$$-1\pi_{TL}+2\pi_{TR}\geq 0 \qquad -1\pi_{TL}+2\pi_{BL}\geq 0$$
$$1\pi_{BL}-2\pi_{BR}\geq 0 \qquad 1\pi_{TR}-2\pi_{BR}\geq 0$$

Correlated-Q Learning
Greenwald/Hall 2003

Four Inequalities

	L	R
1 T	6,6	2,7
B	7,2	0,0

Intuitively, the expected reward to the row player of action T is at least that of action B whenever he in fact plays action T.

$$6\pi_{L|T} + 2\pi_{R|T} \geq 7\pi_{L|T} + 0\pi_{R|T}$$

Correlated-Q Learning
Greenwald/Hall 2003

Four Inequalities

$$6\pi_{L|T} + 2\pi_{R|T} \geq 7\pi_{L|T} + 0\pi_{R|T}$$
$$-1\pi_{L|T} + 2\pi_{R|T} \geq 0$$
$$-1\pi_{TL} + 2\pi_{TR} \geq 0$$

By Bayes Rule:

$$\pi_{xy} = \pi(x|y)\pi(y) = \pi(x \cap y) = \pi(y|x)\pi(x) = \pi_{yx}$$

Four Inequalities

Formula 1

$$6\pi_{L|T} + 2\pi_{R|T} \geq 7\pi_{L|T} + 0\pi_{R|T}$$
$$-1\pi_{L|T} + 2\pi_{R|T} \geq 0$$
$$-1\pi_{TL} + 2\pi_{TR} \geq 0$$

Formula 2

$$7\pi_{L|B} + 0\pi_{R|B} \geq 6\pi_{L|B} + 2\pi_{R|B}$$
$$1\pi_{L|B} - 2\pi_{R|B} \geq 0$$
$$1\pi_{BL} - 2\pi_{BR} \geq 0$$

Formula 3

$$6\pi_{T|L} + 2\pi_{B|L} \geq 7\pi_{T|L} + 0\pi_{B|L}$$
$$-1\pi_{T|L} + 2\pi_{B|L} \geq 0$$
$$-1\pi_{TL} + 2\pi_{BL} \geq 0$$

Formula 4

$$7\pi_{T|R} + 0\pi_{B|R} \geq 6\pi_{T|R} + 2\pi_{B|R}$$
$$1\pi_{T|R} - 2\pi_{B|R} \geq 0$$
$$1\pi_{TR} - 2\pi_{BR} \geq 0$$

note: $\pi_{xy} = \pi(x|y)\pi(y) = \pi(x \cap y) = \pi(y|x)\pi(x) = \pi_{yx}$

Four Inequalities, four variables: it is party time.

$$\pi_{TL} + \pi_{TR} + \pi_{BL} + \pi_{BR} = 1$$

s.t.

$$-1\pi_{TL} + 2\pi_{TR} \geq 0$$
$$1\pi_{BL} - 2\pi_{BR} \geq 0$$
$$-1\pi_{TL} + 2\pi_{BL} \geq 0$$
$$1\pi_{TR} - 2\pi_{BR} \geq 0$$

$$x_1 = \pi_{TL},\ x_2 = \pi_{TR},\ x_3 = \pi_{BL},\ x_4 = \pi_{BR}$$

"Standard Form"?

$$1x_1 + 1x_2 + 1x_3 + 1x_4 = 1$$

s.t.

$$-1x_1 + 2x_2 \geq 0$$

$$1x_3 - 2x_4 \geq 0$$

$$-1x_1 + 2x_3 \geq 0$$

$$1x_2 - 2x_4 \geq 0$$

"Standard Form"?

$$1x_1 + 1x_2 + 1x_3 + 1x_4 = 1$$

s.t.

$$\begin{aligned} 1x_1 - 2x_2 &\leq 0 \\ -1x_3 + 2x_4 &\leq 0 \\ 1x_1 - 2x_3 &\leq 0 \\ -1x_2 + 2x_4 &\leq 0 \end{aligned}$$

New parameters

$$\underbrace{1x_1 + 1x_2 + 1x_3 + 1x_4}_{A} = \underbrace{1}_{b}$$

s.t.

$$\begin{aligned} 1x_1 - 2x_2 &\leq 0 \\ -1x_3 + 2x_4 &\leq 0 \\ 1x_1 - 2x_3 &\leq 0 \\ -1x_2 + 2x_4 &\leq 0 \end{aligned}$$

Old parameters

$$1x_1 + 1x_2 + 1x_3 + 1x_4 = 1$$

s.t.

$$
\begin{array}{rcrcrcrcl}
G & & G & & G & & G & & h \\
1x_1 & - & 2x_2 & + & 0x_3 & + & 0x_4 & \leq & 0 \\
0x_1 & + & 0x_2 & - & 1x_3 & + & 2x_4 & \leq & 0 \\
1x_1 & + & 0x_2 & - & 2x_3 & + & 0x_4 & \leq & 0 \\
0x_1 & - & 1x_2 & + & 0x_3 & + & 2x_4 & \leq & 0
\end{array}
$$

Code

```
from cvxopt import matrix, solvers


c = matrix([1., 1., 1., 1.])
G = matrix([[1., 0., 1., 0.], [-2., 0., 0., -1.],
            [0., -1., -2., 0.], [0., 2., 0., 2.]])
h = matrix([0., 0., 0., 0.])

A = matrix([[1.], [1.], [1.], [1.]])
b = matrix(1.)

solution = solvers.lp(c, G, h, A, b)
```

https://github.com/axonal/cvxopt-tutorial/blob/master/chicken.py

Presentation and Code samples can be found at:
https://github.com/axonal/cvxopt-tutorial