Networked Life

- **Q1:** What makes CDMA work for my smartphone?

Key concepts

- How to use efficiently wireless resources
  - Cellular systems (reuse)
  - TDMA/FDMA, CDMA (sharing)
- Decentralized control
  - Selfish behaviour (games)
  - Stability (convergence of iterations)

Modules

- Wireless communications and CDMA
- The social optimization problem
- Distributed Power Control
- Modelling selfishness: games
- Proving that equilibrium is social optimum

Tutorial

- Examples for CDMA, game theory recap, example for feasible powers

Wireless communications

What are the issues?

What are the issues?

- Interference
- Distance

What are the solutions?

- Reduce "transmission power"
  - Allow only "local" discussions
- Use "orthogonal transmissions" between people
  - use different time slots to talk: TDMA
  - use different voice pitch: FDMA
  - use different language: CDMA

orthogonal: don't interfere

What are the solutions?

- Reduce “transmission power”
  - Allow only “local” discussions
- Use “orthogonal transmissions” between people
  - use different time slots to talk: TDMA
  - use different voice pitch: FDMA
  - use different language: CDMA

Mobile stations communicate through a base station that is near, use channels that are orthogonal

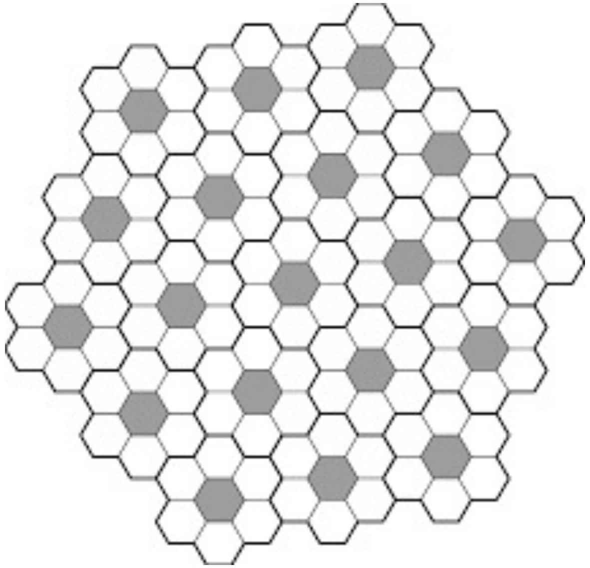
Frequency reuse: cellular systems

- Suppose we want to allocate 100 channels to each cell. How many in total I need to serve this area?

suppose n orth. channels are enough to serve any cell

Frequency reuse: cellular systems

- But we may have to **reuse channels**
- How do we solve the interference problem for stations that might use the same channel?

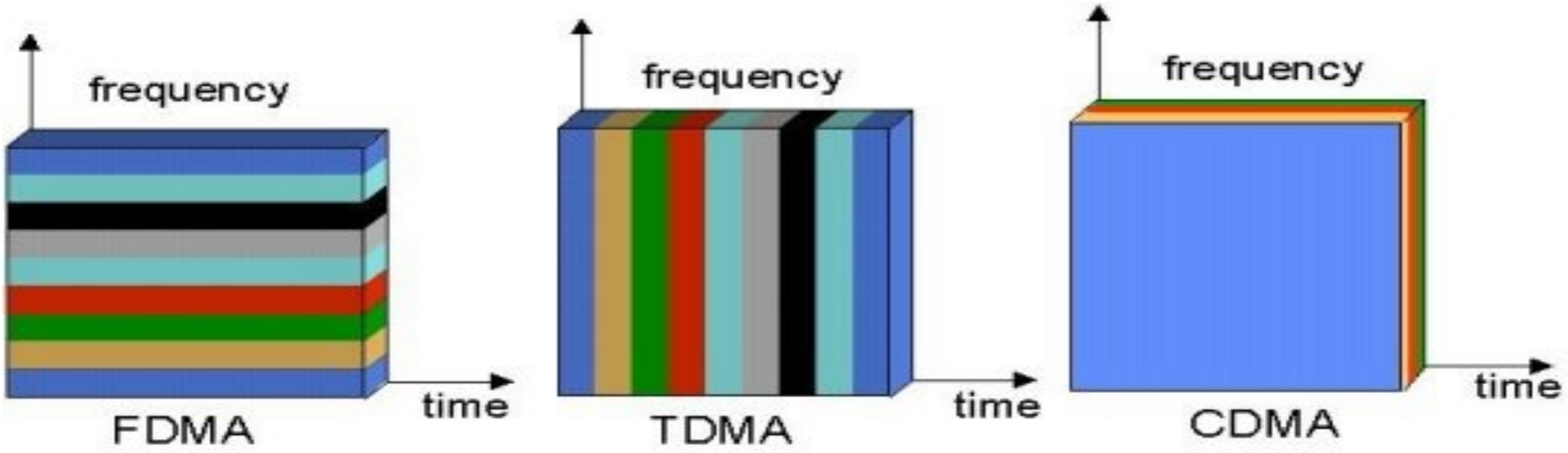


suppose n orth. channels are enough to serve any cell

We can **reuse** these n channels in each cell numbered i because the cells with same number are far apart

We need a total of $n \times 7$ orthogonal channels independently of total area

Orthogonal channels: FDAM, TDMA, CDMA



CDMA: each channel uses all frequencies during all times

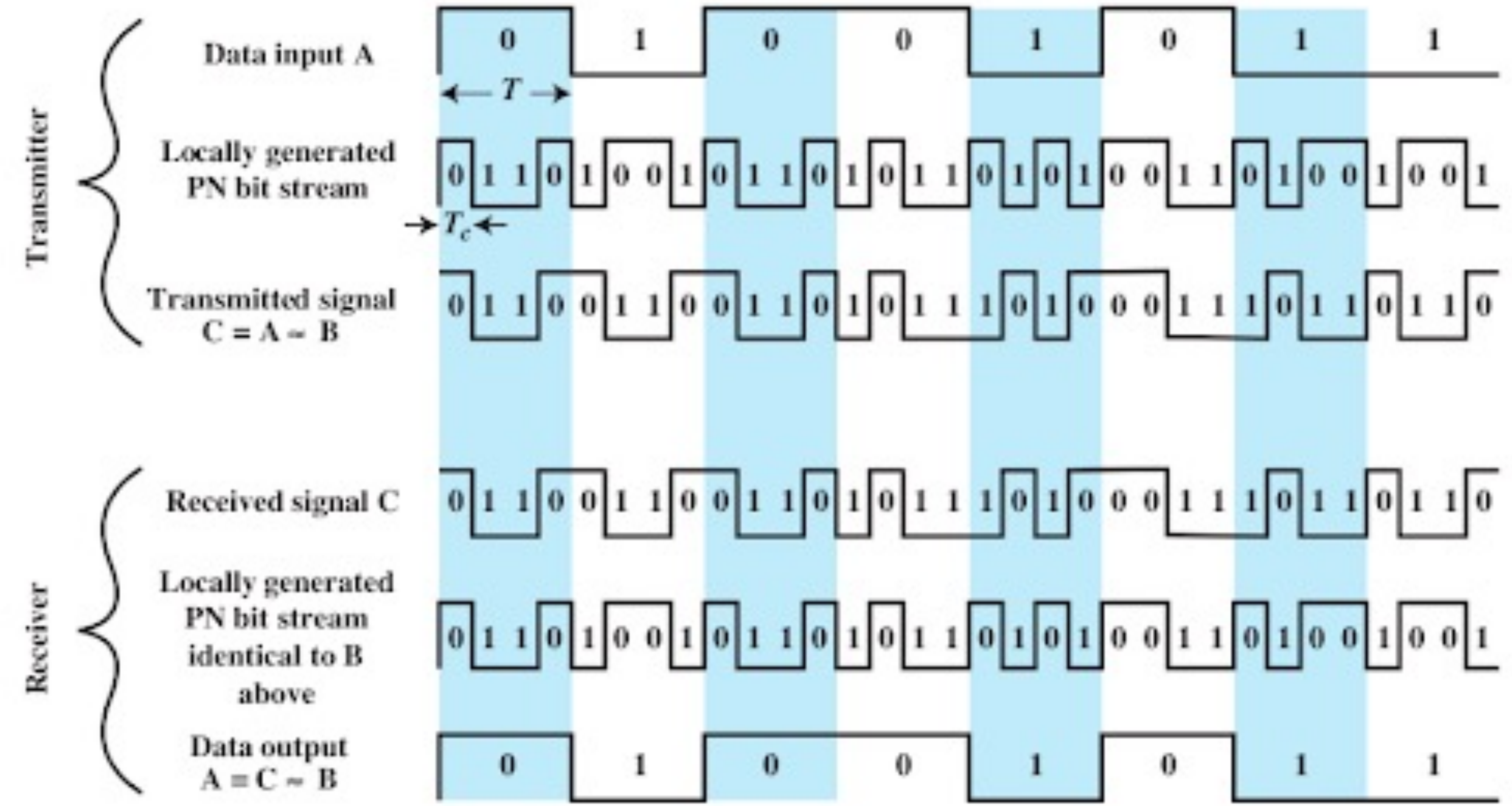
Channel: uses a distinct pseudo-random code

Problem: channel interference!

What you need to know about CDMA

- Each channel sender uses a specific pseudo-random code (“spreading code”) to encode its signal
- Each channel receiver uses the same code to decode the total received signal:
  - Recovers exactly the signal sent by the sender
  - All signals from other channels, when decoded, **appear like noise (“interference”)**
  - For more details see Wikipedia https://en.wikipedia.org/wiki/Code-division_multiple_access
- Summary: a receiver receives (useful) **signal +** (not useful) **noise**

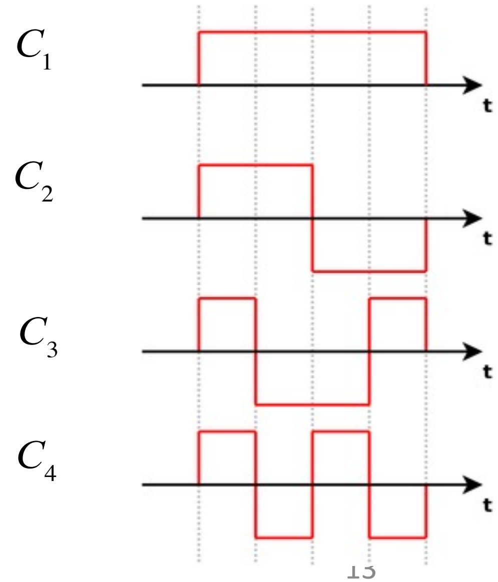
Example



we encode bits as voltage values:
bit 0: +1v
bit 1: -1v

pointwise product of code×data

inner product ⊗ of code ⊗ data



Key question

- How do we analyze systems with *negative externalities*?
- A user’s transmission adds noise to the channels of other users

The optimization problem formulation

Channel gain

$$G_{ij} = \text{gain in receiver } i \text{ from sender } j$$

Receiver i receives
G_{ij} units of power

Sender j transmits
1 unit of power

$$i \xleftarrow{\;G_{ij}\;} j$$

" to i from j "

- A set of senders $S = \{s_1, \dots, s_n\}$ and receivers $R = \{r_1, \dots, r_n\}$
- A pair $\{r_i, s_j\}$ is referred to as "channel (ij)"
- Channel $i \triangleq$ channel (ii)

Interference

- Suppose every sender i is paired with a receiver i
- CDMA channel i: pair of (sender i, receiver i) = channel (ii)
- Each CDMA channel i cares for its signal reception quality: signal to interference ratio

$$SIR_i = \frac{G_{ii}p_i}{\sum_{j \neq i} G_{ij}p_j + n_i}$$

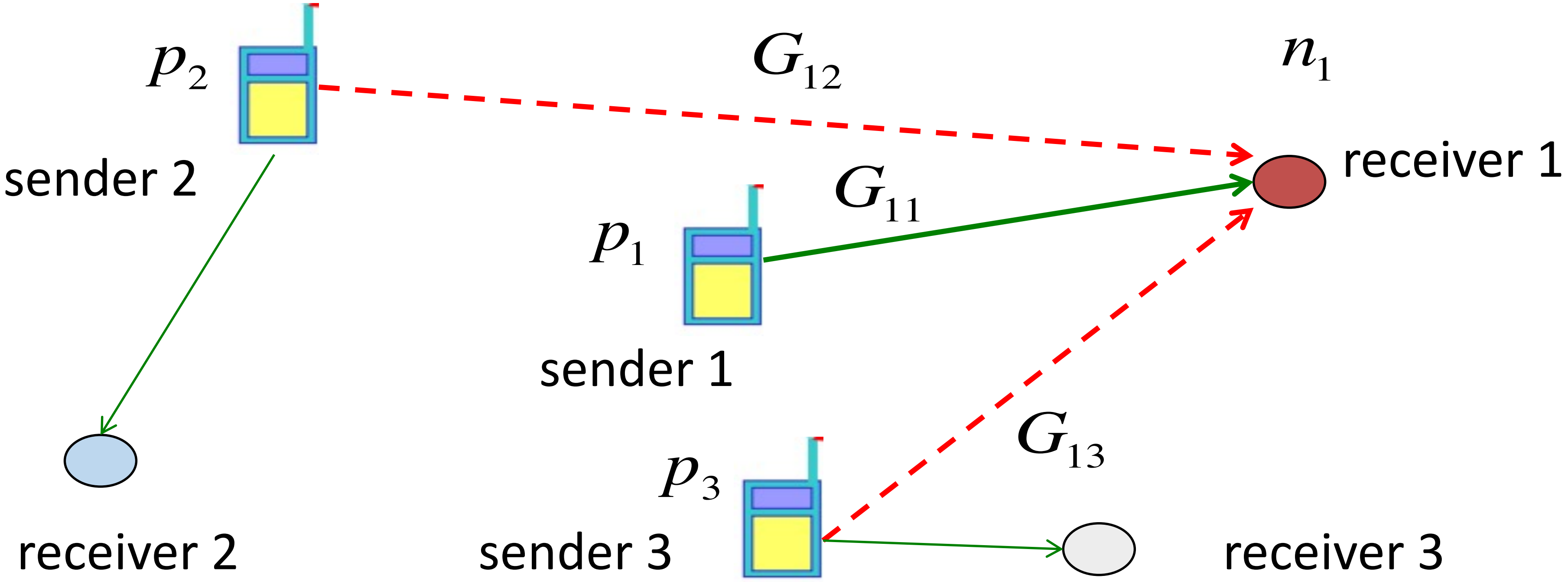
- Channel capacity C: bits/sec

$$C = B\log_2(1 + \frac{S}{N}) = B\log_2(1 + SIR)$$

B: width of frequency band

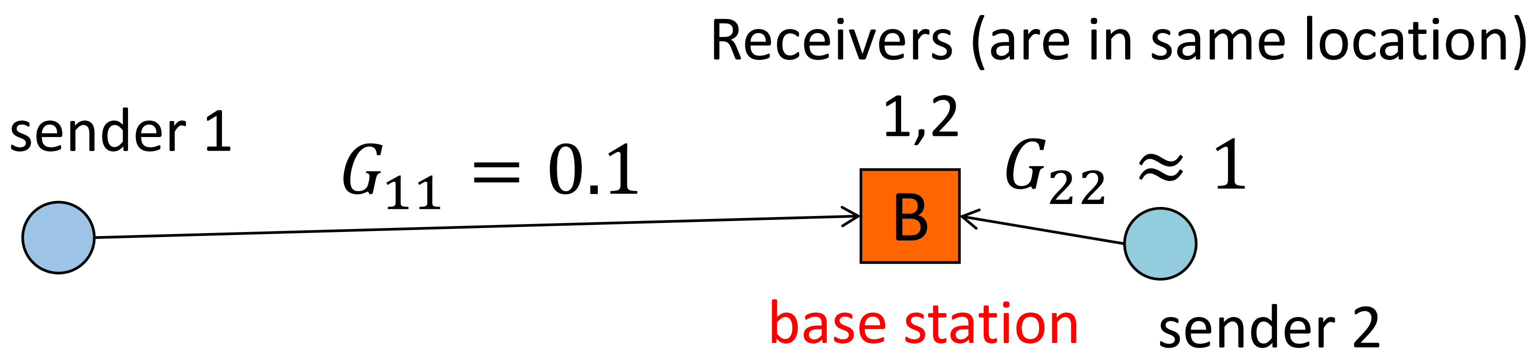
Interference example

$$SIR_1 = \frac{G_{11}p_1}{G_{12}p_2 + G_{13}p_3 + n_1} \qquad channel_i = (sender_i, receiver_i)$$



The near-far problem at a base station

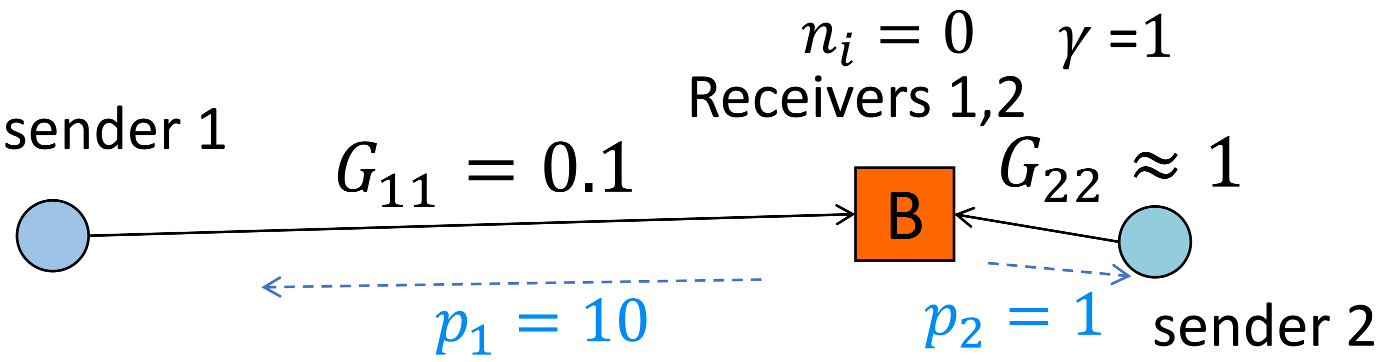
- Can we have senders transmit at predetermined constant power?
- E.g., $p_i = 1, n_i = 0, i = 1,2$



- $SIR_1 = \frac{0.1}{1} = 0.1, SIR_2 = \frac{1}{0.1} = 10$
- Is it fair? How to improve? need "power control"

Power control: centralized vs. distributed

- Example of simple fair policy: achieve the same SIR=γ at the receiver



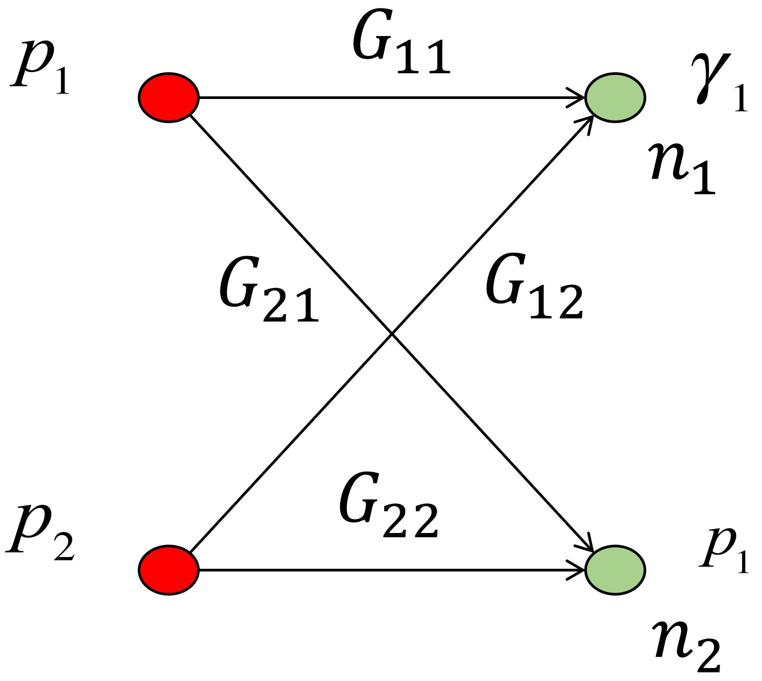
All receivers are in same location

$$SIR_i = \frac{G_{ii}p_i}{G_{jj}p_j + 0} = \frac{1}{1} = 1$$

- **Centralized**: the receivers dictate to the senders the power to send
- **Distributed** algorithm: Each receiver signals its sender to increase or decrease
  - This is Distributed Power Control! (see later)
  - It takes many steps to reach the desired operating point (need to converge)

Centralized solution for fixed *SIRs*

- Assume full information at the central system manager (central planner)
- Find the transmit powers p_i to get a target $SIR_i = \gamma_i$ in the case $G_{ii} = 0.8, G_{ij} = 0.4, \gamma_1 = 1, \gamma_2 = 2, n_1 = n_2 = 0.5$



2 equation in 2 unknowns p_1, p_2

$$SIR_1 = \frac{G_{11}p_1}{G_{12}p_2+n_1},\ SIR_2 = \frac{G_{22}p_2}{G_{21}p_1+n_2}$$

We want $SIR_1 = \gamma_1, SIR_2 = \gamma_2$

- Distributed: can we find the solution without full information?

More generally: feasible power set

- Given the gains of the system, noises, and minimum values γ_i for the SIR_is, find $p_1, \dots, p_n$ s.t. feasible, i.e.,

$$SIR_i = \frac{G_{ii}p_i}{\sum_{j \neq i} G_{ij}p_j + n_i} \geq \gamma_i, \qquad \forall i = 1, \dots, n$$

- Optimization problem: Find feasible powers that minimize $\sum_i p_i$

General form of constraints

$$\frac{p_i G_{ii}}{\sum_{j \neq i} p_j G_{ij} + n_i} \geq \gamma_i \Leftrightarrow p_i \geq \frac{\gamma_i}{G_{ii}} \left(\sum_{j \neq i} p_j G_{ij} + n_i \right)$$

$$\begin{pmatrix} p_1 \\ \dots \\ p_i \\ \dots \\ p_n \end{pmatrix} \geq \begin{pmatrix} \gamma_1 & & & & 0 \\ & \dots & & & \\ & & \gamma_i & & \\ & & & \dots & \\ 0 & & & & \gamma_n \end{pmatrix} \begin{pmatrix} 0 & \frac{G_{12}}{G_{11}} & \dots & & \dots & & \frac{G_{1n}}{G_{11}} \\ \dots & \dots & \dots & & \dots & & \dots \\ \frac{G_{i1}}{G_{ii}} & \frac{G_{i2}}{G_{ii}} & \dots & 0 & \frac{G_{ii+1}}{G_{ii}} & \dots & \frac{G_{in}}{G_{ii}} \\ \dots & \dots & \dots & & \dots & & \dots \\ \frac{G_{n1}}{G_{nn}} & \dots & \dots & & \dots & & 0 \end{pmatrix} \begin{pmatrix} p_1 \\ \dots \\ p_i \\ \dots \\ p_n \end{pmatrix} + \begin{pmatrix} \frac{\gamma_1 n_1}{G_{11}} \\ \dots \\ \frac{\gamma_i n_i}{G_{ii}} \\ \dots \\ \frac{\gamma_n n_n}{G_{nn}} \end{pmatrix}$$

$$p \geq D(\gamma)Fp + v \Leftrightarrow (I - D(\gamma)F)p \geq v$$

The social optimization problem

- Find the transmit powers that achieve for every MS the desired SIR with the minimum total social cost (in our case the total battery power)

$$p^* \triangleq \min 1^T p \quad s.t. \ (I - D(\gamma)F)p \geq v$$

Socio-economic problem:

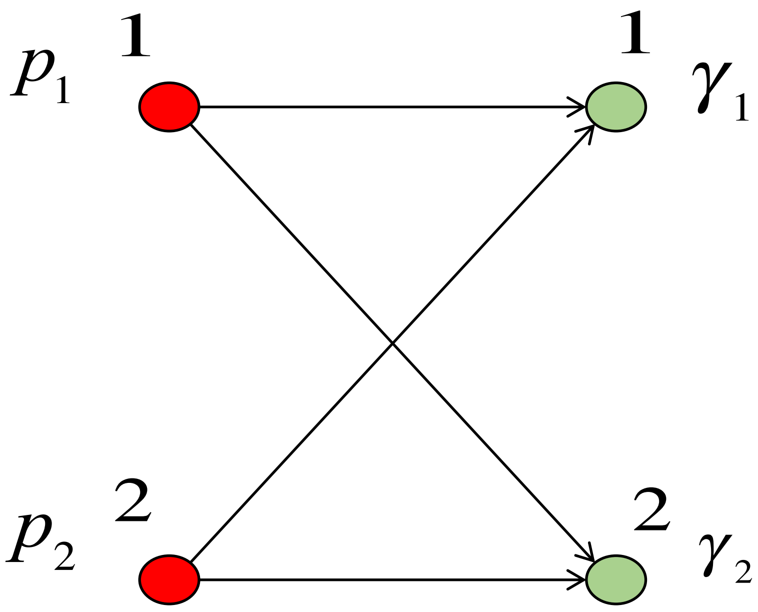
- Each user gets **value** from the system by **hurting others**
- She incurs some **cost** for her actions (e.g., battery power)

Two system views:

- **Centralized**: Suppose a “Social Planner” has full information, solves the problem centrally, and **dictates** to each user its action. What is the **socially optimal** set of actions? (i.e., we do **optimization**)
- **Distributed:** What happens if each user acts selfishly? (i.e., we analyze a **game**)
  - Will users follow the suggestion of the Social Planner?

Example for $n = 2$

- Given the gains of the system, noises, and minimum values γ_i for SIR_i, are there *feasible* transmit powers? i.e., satisfy $SIR_i \geq \gamma_i$

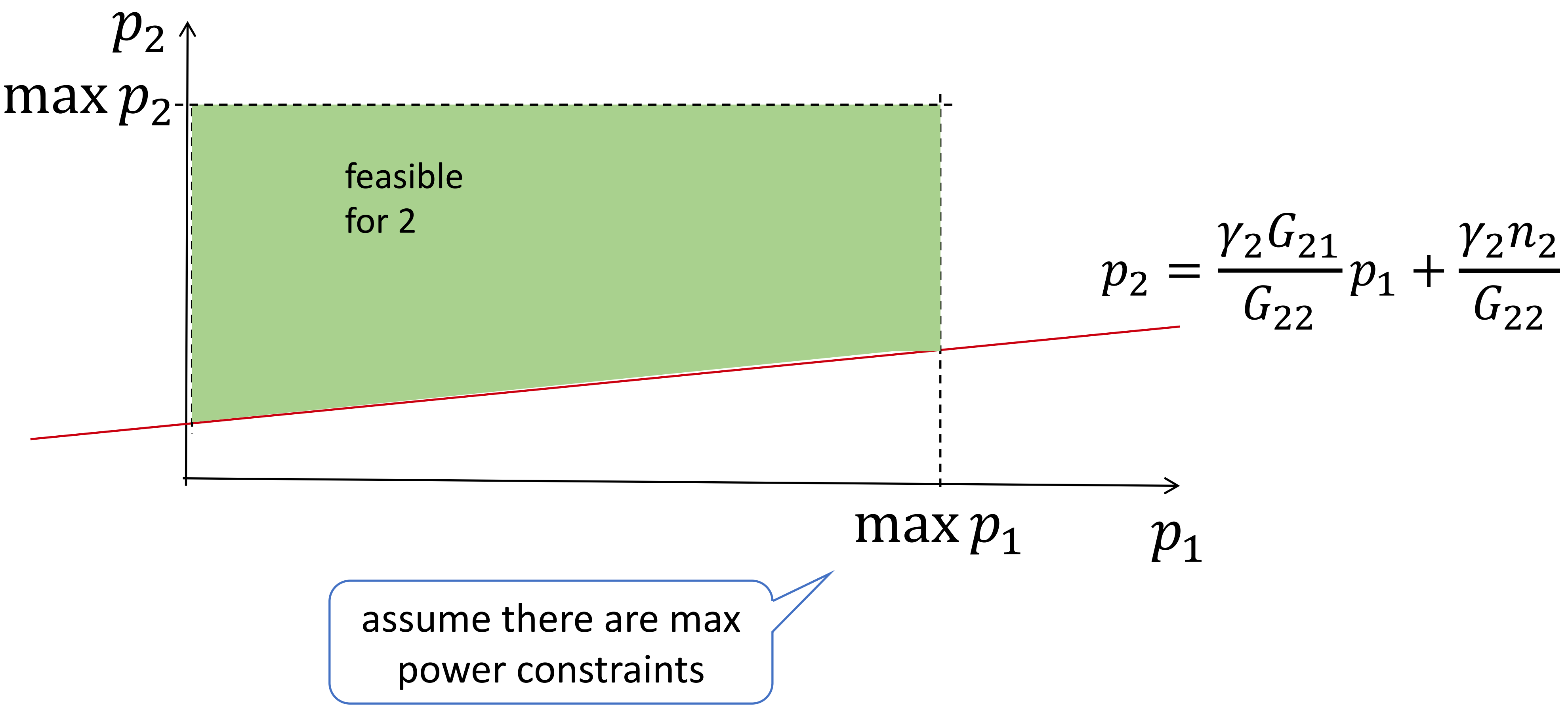


$$SIR_1 = \frac{G_{11}p_1}{G_{12}p_2+n_1}, \quad SIR_2 = \frac{G_{22}p_2}{G_{21}p_1+n_2}$$

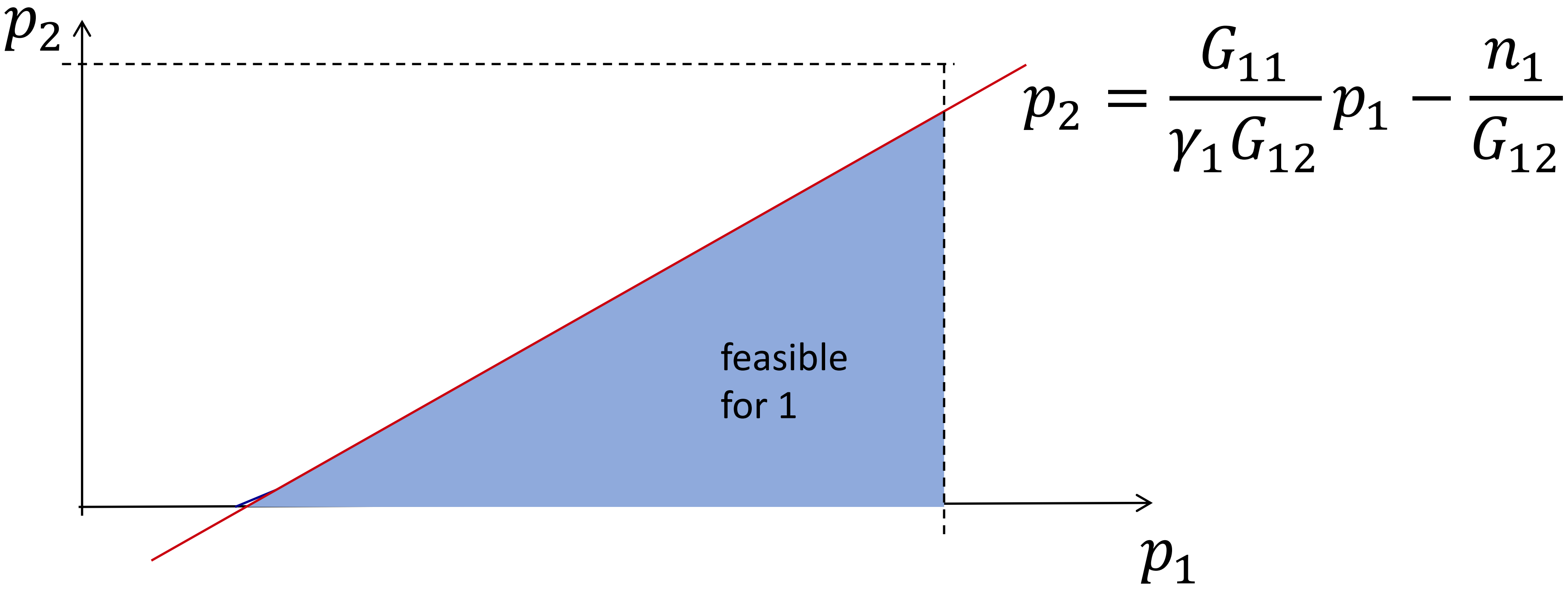
We want $SIR_1 \geq \gamma_1, SIR_2 \geq \gamma_2$

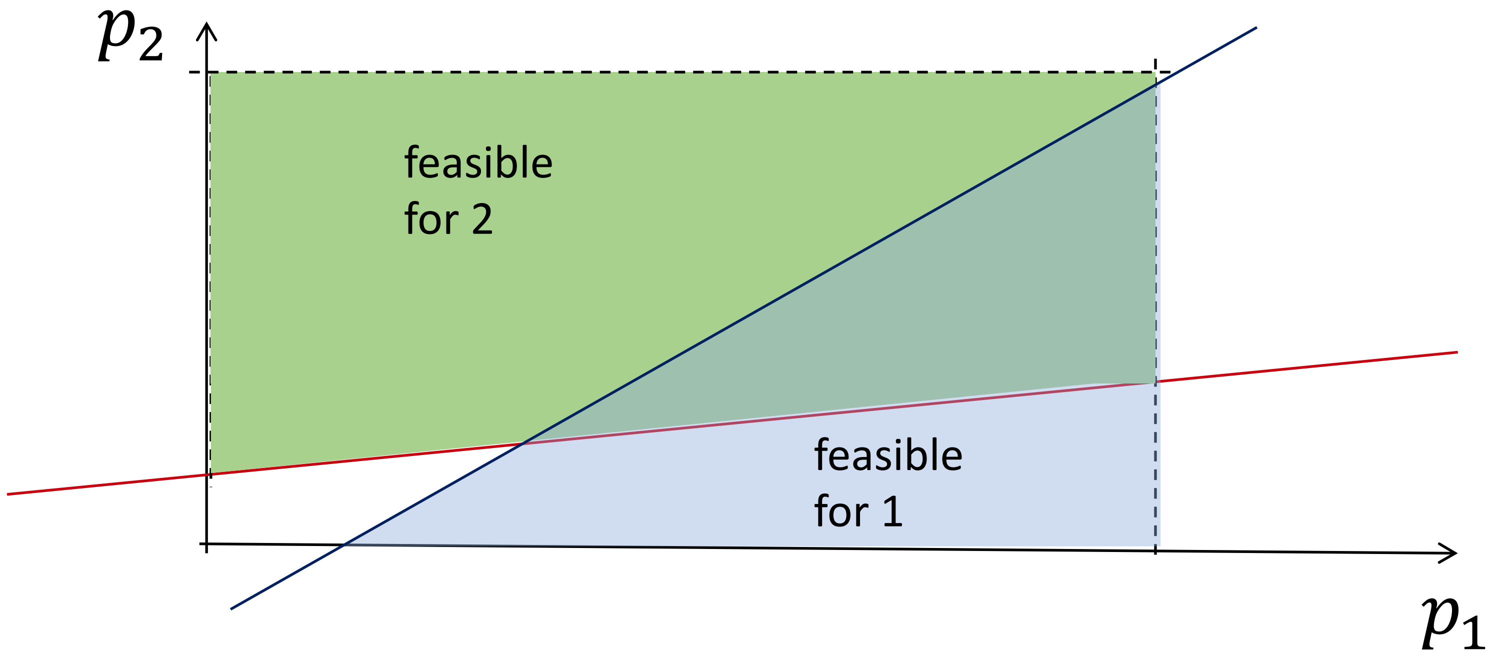
- Conditions for a solution?
- Can we find it with a distributed algorithm?

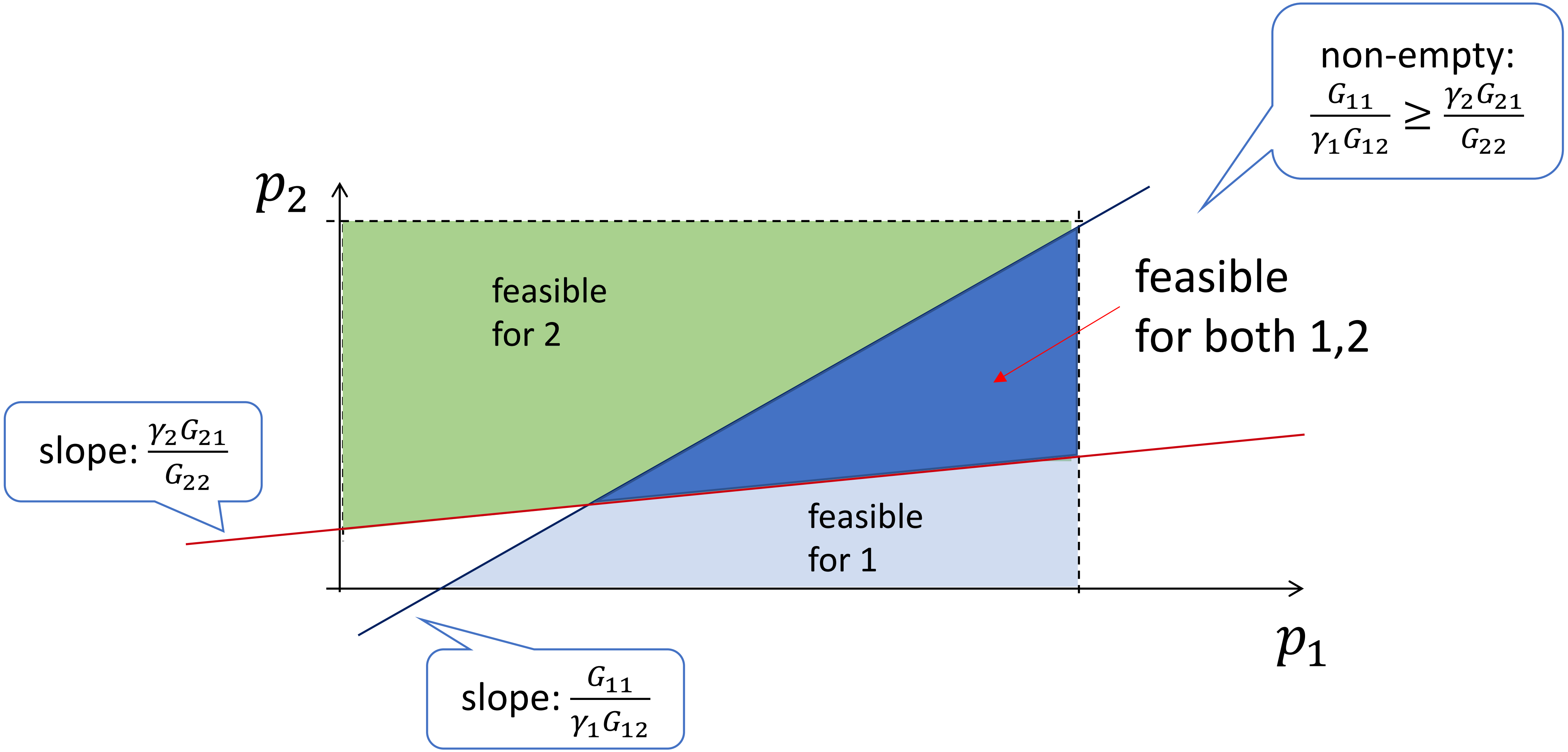
$$\frac{G_{22}p_2}{G_{21}p_1+n_2} \geq \gamma_2 \Rightarrow G_{22}p_2 \geq \gamma_2(G_{21}p_1 + n_2) \Rightarrow p_2 \geq \frac{\gamma_2 G_{21}}{G_{22}}p_1 + \frac{\gamma_2 n_2}{G_{22}}$$



$$\frac{G_{11}p_1}{G_{12}p_2 + n_1} \ge \gamma_1 \Rightarrow G_{12}p_2 + n_1 \le \frac{G_{11}p_1}{\gamma_1} \Rightarrow p_2 \le \frac{G_{11}}{\gamma_1 G_{12}}p_1 - \frac{n_1}{G_{12}}$$




p_2
feasible
for 2
feasible
for 1
p_1

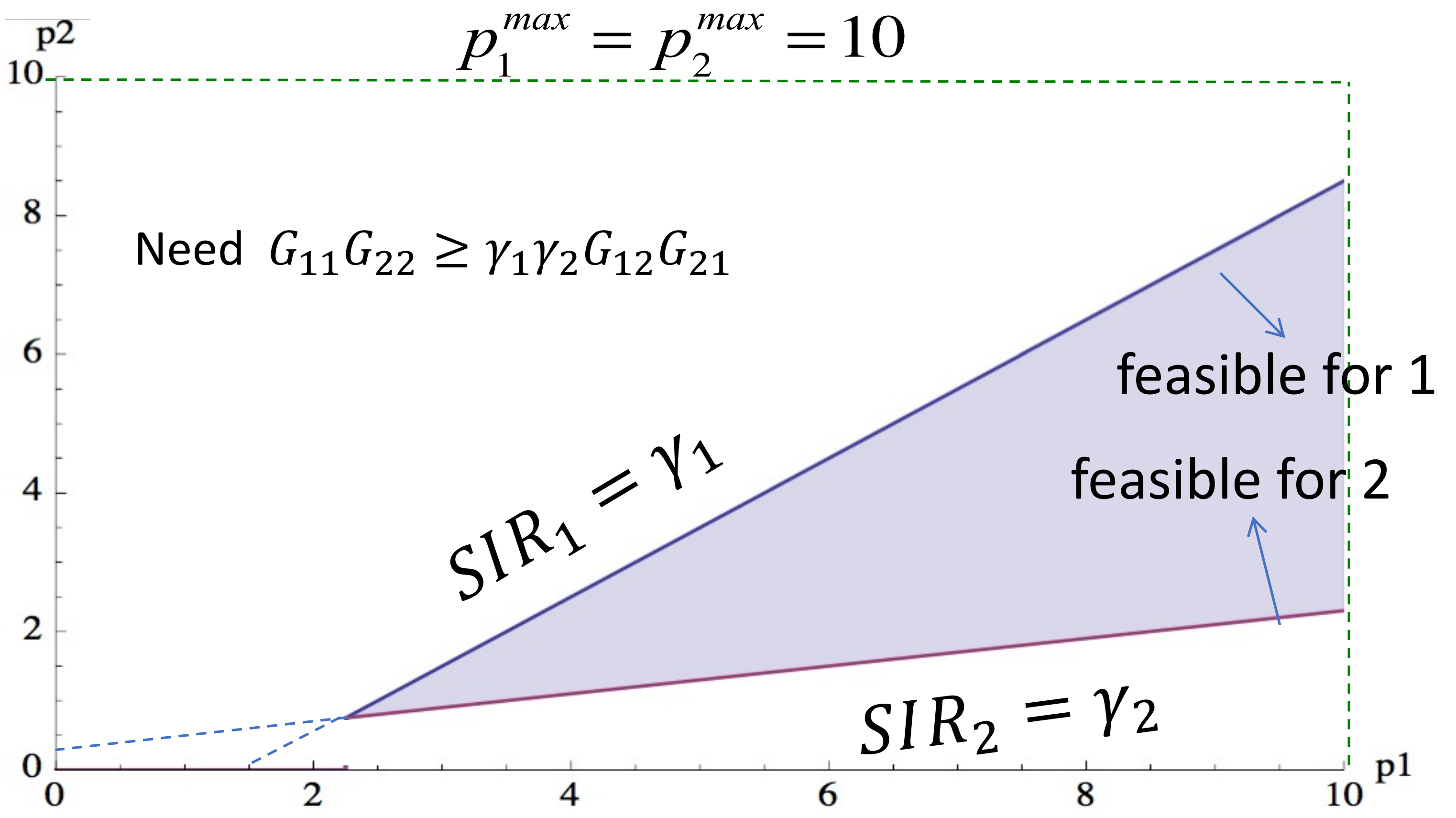


non-empty: $\frac{G_{11}}{\gamma_1 G_{12}} \geq \frac{\gamma_2 G_{21}}{G_{22}}$
p_2
feasible for 2
feasible for both 1,2
slope: $\frac{\gamma_2 G_{21}}{G_{22}}$
feasible for 1
p_1
slope: $\frac{G_{11}}{\gamma_1 G_{12}}$

Numerical example

$$G_{11} = 1,\ G_{22} = 1,$$
$$G_{12} = G_{21} = 0.2$$
$$\gamma_1 = 5, \gamma_2 = 1, n = 0.3$$

$$SIR_i = \frac{G_{ii} p_i}{G_{ij} p_j + n}$$



general condition for the feasible region not to be empty?

DPC: a distributed algorithm for power control

Distributed Power Control (DPC) algorithm

- γ_i= target SIR for channel i
- For all i:

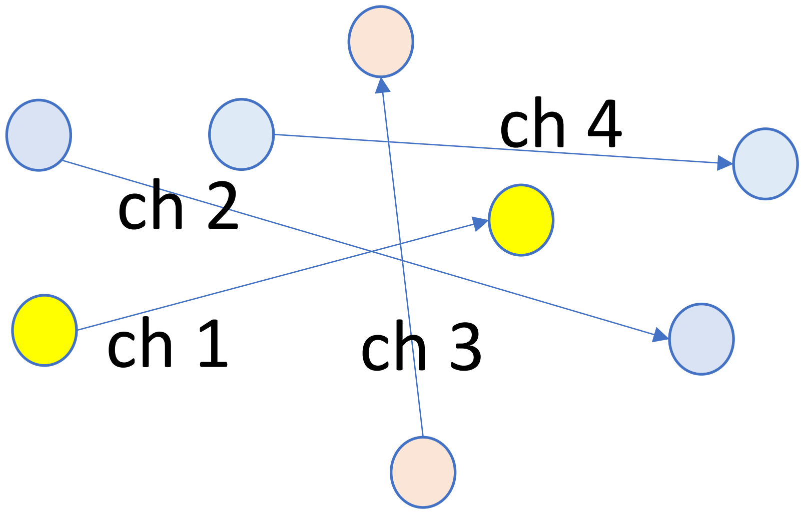
try in one step to get your desired γ_i assuming nothing else changes

$$p_i[t+1] = \frac{\gamma_i}{SIR_i[t]} p_i[t]$$

$$\boxed{SIR_i[t] = \frac{G_{ii} p_i[t]}{\sum_{j \neq i} G_{ij} p_j + n_i}}$$

- Does it converge?
- Does it solve some optimization problem?

Example



Channel gains

Receiving station	Transmitting station 1	2	3	4
1	1	0.1	0.2	0.3
2	0.2	1	0.1	0.1
3	0.2	0.1	1	0.1
4	0.1	0.1	0.3	1

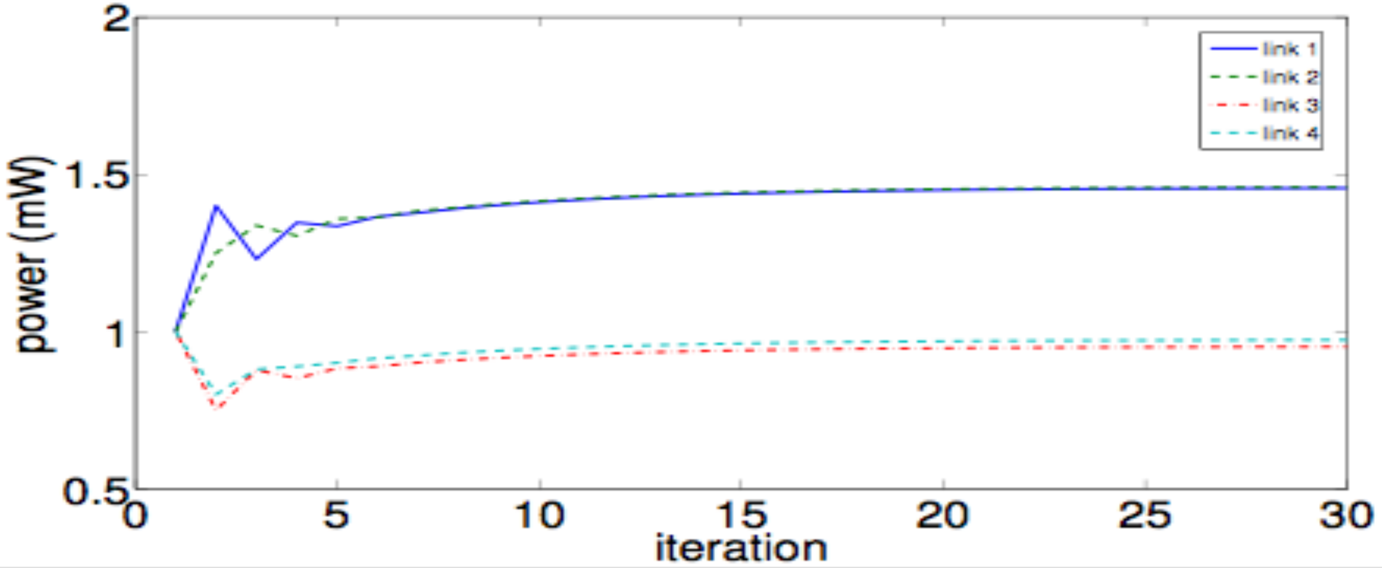
$$\gamma_1 = 2.0$$
$$\gamma_2 = 2.5$$
$$\gamma_3 = 1.5$$
$$\gamma_4 = 2.0.$$

Iterations

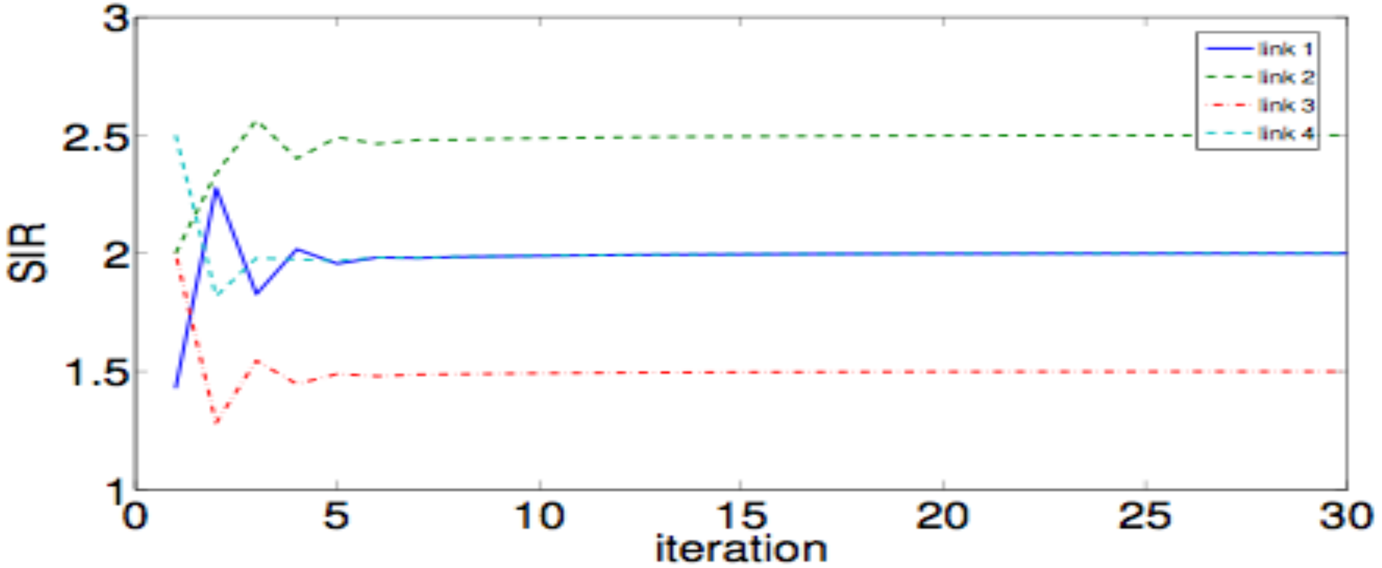
$$p_1[1] = \frac{\gamma_1}{\mathrm{SIR}_1[0]} p_1[0] = \frac{2.0}{1.43} \times 1.0 = 1.40$$
$$p_2[1] = \frac{\gamma_2}{\mathrm{SIR}_2[0]} p_2[0] = \frac{2.5}{2.00} \times 1.0 = 1.25$$
$$p_3[1] = \frac{\gamma_3}{\mathrm{SIR}_3[0]} p_3[0] = \frac{1.5}{2.00} \times 1.0 = 0.75$$
$$p_4[1] = \frac{\gamma_4}{\mathrm{SIR}_4[0]} p_4[0] = \frac{2.0}{1.67} \times 1.0 = 1.20.$$

$$\mathrm{SIR}_1[1] = \frac{1 \times 1.40}{0.1 \times 1.25 + 0.2 \times 0.75 + 0.3 \times 1.20 + 0.1} = 1.90$$
$$\mathrm{SIR}_2[1] = \frac{1 \times 1.25}{0.2 \times 1.40 + 0.1 \times 0.75 + 0.1 \times 1.20 + 0.1} = 2.17$$
$$\mathrm{SIR}_3[1] = \frac{1 \times 0.75}{0.2 \times 1.40 + 0.1 \times 1.25 + 0.1 \times 1.20 + 0.1} = 1.20$$
$$\mathrm{SIR}_4[1] = \frac{1 \times 1.20}{0.1 \times 1.40 + 0.1 \times 1.25 + 0.1 \times 0.75 + 0.1} = 2.03.$$

Iterations



link 1
link 2
link 3
link 4
power (mW)
2
1.5
1
0.5
0
5
10
15
20
25
30
iteration



link 1
link 2
link 3
link 4
SIR
3
2.5
2
1.5
1
0
5
10
15
20
25
30
iteration

Analysis of DPC

Matrix form of DPC

- DPC: $\quad p[t+1] = D(\gamma)Fp[t] + v$
- Why is this correct?

$$\begin{pmatrix} p_1[t+1] \\ p_2[t+1] \\ p_3[t+1] \end{pmatrix} = \begin{pmatrix} \gamma_1 & 0 & 0 \\ 0 & \gamma_2 & 0 \\ 0 & 0 & \gamma_3 \end{pmatrix} \begin{pmatrix} 0 & \frac{G_{12}}{G_{11}} & \frac{G_{13}}{G_{11}} \\ \frac{G_{21}}{G_{22}} & 0 & \frac{G_{23}}{G_{22}} \\ \frac{G_{31}}{G_{33}} & \frac{G_{32}}{G_{33}} & 0 \end{pmatrix} \begin{pmatrix} p_1[t] \\ p_2[t] \\ p_3[t] \end{pmatrix} + \begin{pmatrix} \frac{n_1\gamma_1}{G_{11}} \\ \frac{n_2\gamma_2}{G_{22}} \\ \frac{n_3\gamma_3}{G_{33}} \end{pmatrix}$$

$$\begin{pmatrix} p_1[t+1] \\ p_2[t+1] \\ p_3[t+1] \end{pmatrix} = \begin{pmatrix} \gamma_1 & 0 & 0 \\ 0 & \gamma_2 & 0 \\ 0 & 0 & \gamma_3 \end{pmatrix} \begin{pmatrix} 0 & \frac{G_{12}}{G_{11}} & \frac{G_{13}}{G_{11}} \\ \frac{G_{21}}{G_{22}} & 0 & \frac{G_{23}}{G_{22}} \\ \frac{G_{31}}{G_{33}} & \frac{G_{32}}{G_{33}} & 0 \end{pmatrix} \begin{pmatrix} p_1[t] \\ p_2[t] \\ p_3[t] \end{pmatrix} + \begin{pmatrix} \frac{n_1\gamma_1}{G_{11}} \\ \frac{n_2\gamma_2}{G_{22}} \\ \frac{n_3\gamma_3}{G_{33}} \end{pmatrix}$$

$$\begin{aligned} p_1[t+1] &= \gamma_1 (p_2[t]\frac{G_{12}}{G_{11}} + p_3[t]\frac{G_{13}}{G_{11}}) + \gamma_1 \frac{n}{G_{11}} \\ &= \gamma_1 \frac{p_2[t]G_{12} + p_3[t]G_{13} + n_1}{G_{11}} \\ &= \gamma_1 \frac{p_2[t]G_{12} + p_3[t]G_{13} + n_1}{p_1[t]G_{11}} p_1[t] \\ &= \gamma_1 \frac{1}{\frac{p_1[t]G_{11}}{p_2[t]G_{12}+p_3[t]G_{13}+n_1}} p_1[t] \\ &= \frac{\gamma_1}{SNR_1[t]} p_1[t] \end{aligned}$$

DPC:

Analysis of DPC

$$p[t+1] = D(\gamma)F\,p[t] + v$$

$$p[0]$$

$$p[1] = DF\,p[0] + v$$

$$p[2] = DF\,p[1] + v = (DF)^2 p[0] + (DF + I)v$$

$$p[3] = DF\,p[2] + v = (DF)^3 p[0] + ((DF)^2 + DF + I)v$$

$$p[4] = DF\,p[3] + v = \ \ldots\ = (DF)^4 p[0] + ((DF)^3 + (DF)^2 + DF + I)v$$

$$p[n] = (DF)^n p[0] + ((DF)^{n-1} + (DF)^{n-2} + \cdots + DF + I)v$$

Why converge?

Analysis of DPC

$$p[n] = (DF)^n p[0] + ((DF)^{n-1} + (DF)^{n-2} + \cdots + DF + I)v$$

Why converge?

If the spectral radius (abs. value of largest eigenvalue): $\rho(DF) < 1 \Rightarrow$

1) $(DF)^n \to 0$ and

2) $(DF)^{n-1} + (DF)^{n-2} + \cdots + DF + I \to (\mathrm{I} - DF)^{-1}$

In analogy to

$x < 1 \Rightarrow x^n \to 0$ and $x^{n-1} + x^{n-2} + \cdots + x + 1 \to \frac{1}{1-x} = (1-x)^{-1}$

Analysis of DPC

$$p[n] = (DF)^n p[0] + ((DF)^{n-1} + (DF)^{n-2} + \cdots + DF + I)v$$

Why converge?

If the spectral radius (largest eigenvalue): $\rho(DF) < 1 \Rightarrow$

$$p[n] \to 0 \cdot p[0] + \left(\sum_{k=0}^{\infty} (DF)^k\right) v$$

$$= 0 + (I - DF)^{-1} v$$

$$= (I - DF)^{-1} v$$

Big picture summary

Power management for mobile stations:

1. Social optimization: centralized opt. problem (social optimum SO)
2. Selfish behaviour: solution of game (Nash Equilibrium NE), best-response
3. DPC: a sensible engineering proposal

Issues?

- What is the social optimum (SO) solution? Use as benchmark
- How close is the NE of the game?
- How good is the solution of DPC?

DPC equilibrium = NE = SO
In general NE<SO

Modelling the DPC as a game

What is a game

- A game is huge abstraction of reality
- 3-tuple definition:
  - Players
  - Strategy space per player
  - Payoff function per player
- The most commonly used solution concepts are equilibrium concepts, most famously the Nash equilibrium

Example: Prisoner's dilemma

- Game: Player A, Player B,   Strategies: {*deny, confess*}
- Payoff matrix: $U_A(a,b), U_B(a,b)$
- Game description: strategic form, extensive form (dynamics)

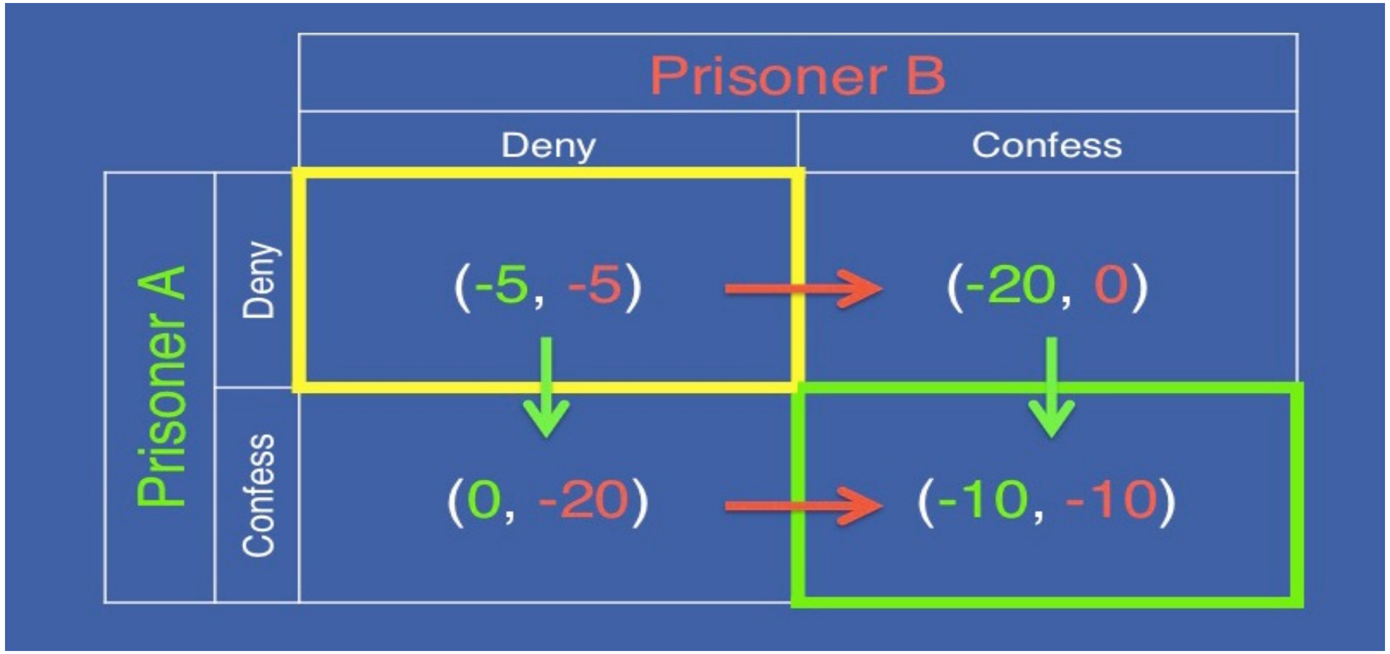
Simultaneous games
Sequential games

		Prisoner B	
		Deny	Confess
Prisoner A	Deny	(-5, -5)	
	Confess		

$$U_A(deny, deny) = -5$$
$$U_B(deny, deny) = -5$$

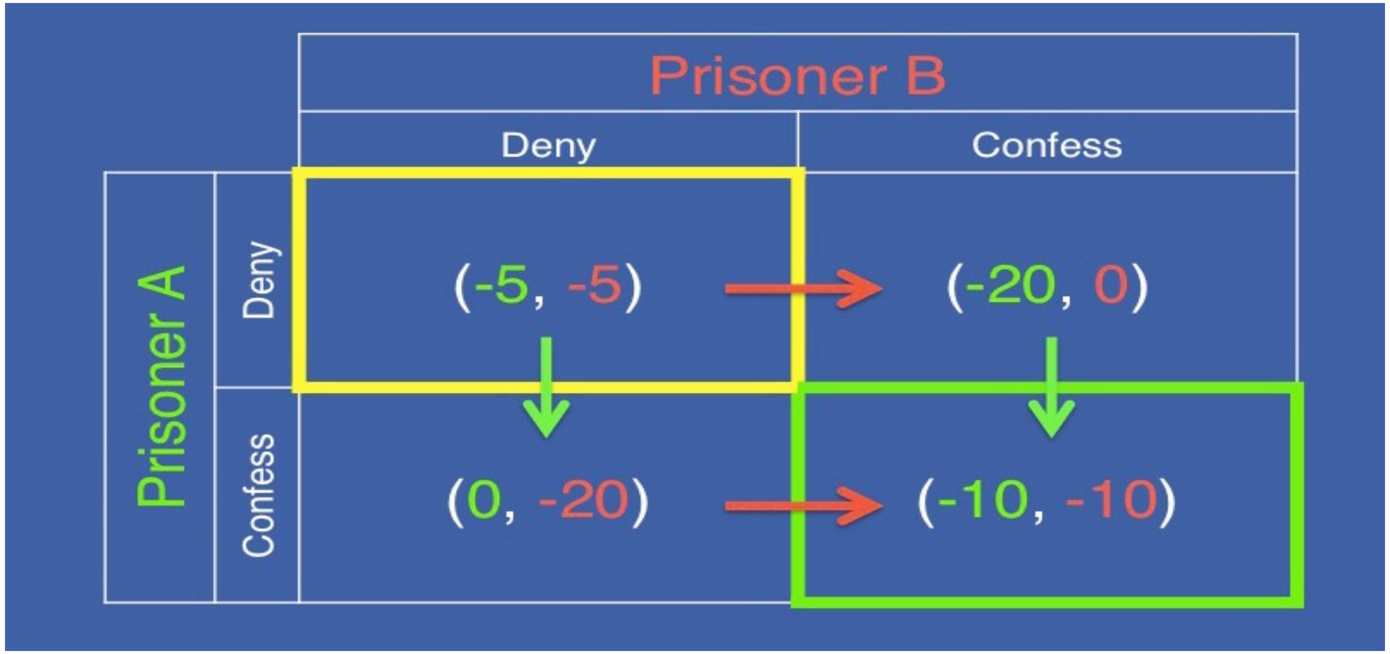
Example: Prisoner's dilemma

- Game: Player A, Player B,   Strategies: {*deny, confess*}
- Best response strategies: $s \in BR_A(.\,, b)$ iff $U_A(s, b) \geq U_A(s', b)$ for all $s' \in S_A$



Example: Prisoner's dilemma

- Game: Player A, Player B,   Strategies: {*deny, confess*}
- Best response strategies: $s \in BR_A(.,b)$ iff $U_A(s,b) \geq U_A(s',b)$ for all $s' \in S_A$



Nash Equilibrium NE:
(a, b) is a NE iff

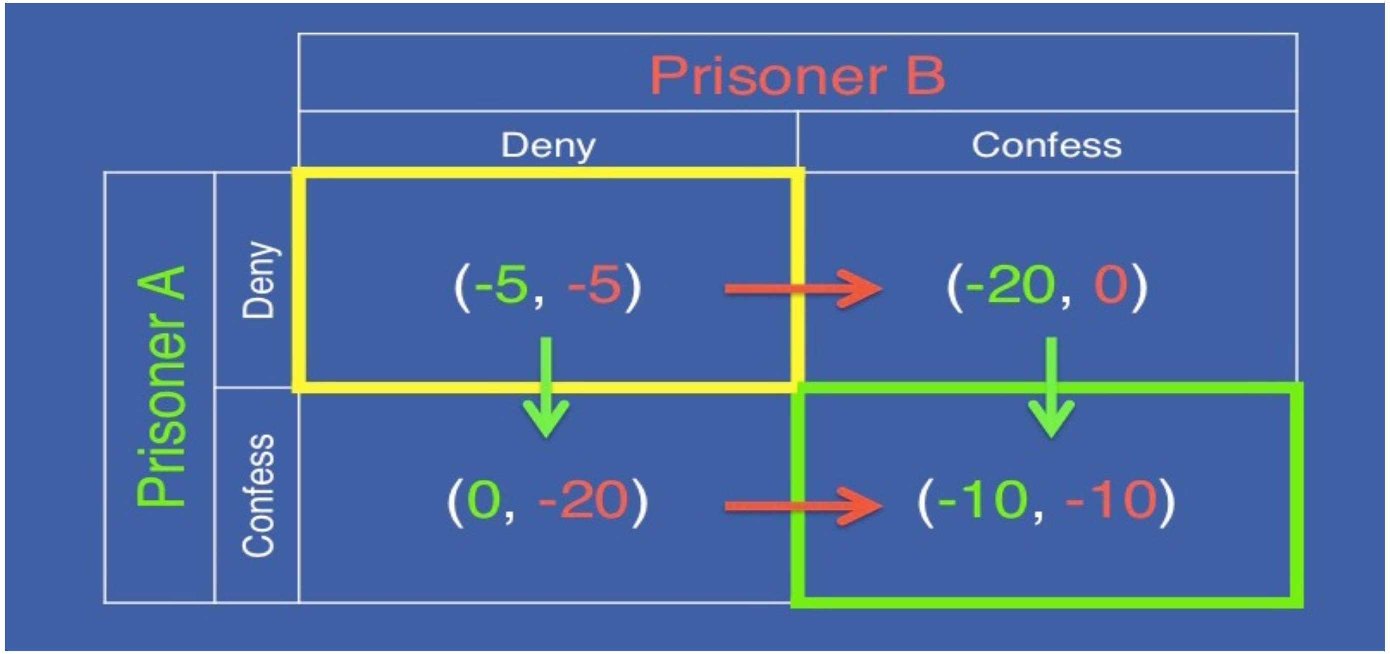
$$a \in BR_A(.,b),$$
$$b \in BR_B(a,.)$$

The most commonly used solution concept for a game

may apply a **refinement** to narrow down the solutions

Example: Prisoner's dilemma

- Game: Player A, Player B, Strategies: {*deny, confess*}
- Dominant strategy: $s \in D_{\mathrm{A}}$ iff $U_A(s,b) \geq U_A(s',b)$ for all $s' \in S_A$, $b \in S_B$

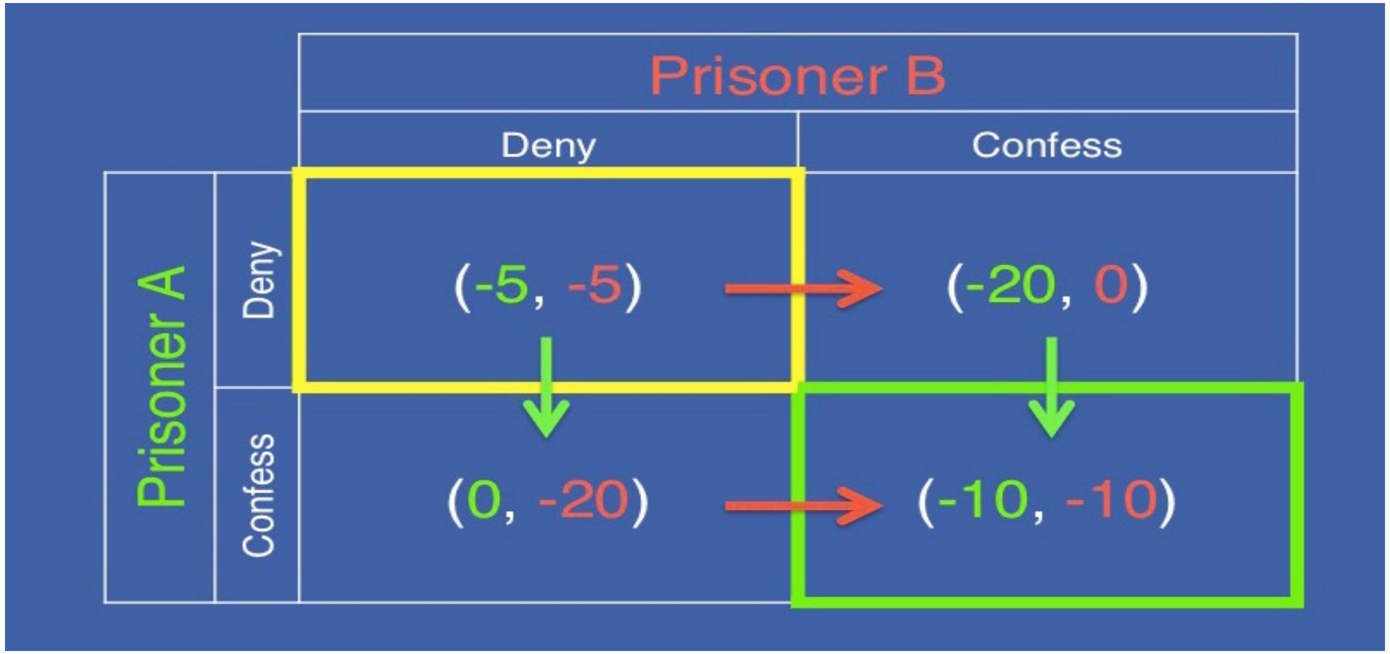


$\textit{confess} \in D_A$,
$\textit{confess} \in D_B$

Solution: (confess, confess)
We eliminate all strictly dominated strategies

Example: Prisoner's dilemma

- Game: Player A, Player B,   Strategies: {*deny, confess*}
- Socially optimal strategies: maximize $U_A + U_B$
- Pareto optimal strategies



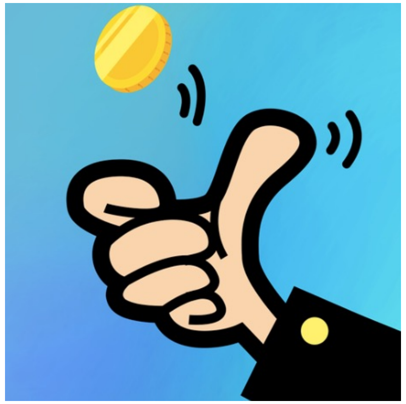
Example: the hawk – dove game

	Player 2		
		Compromise	Don't compromise
Player 1	Compromise	(0,0)	(-1,1)
	Don't compromise	(1,-1)	(-10,-10)

Find the Nash Equilibria in pure strategies

Mixed strategies

Generalized coins: many faces



- Flip a (generalized) coin to pick my action
- My strategy = **type of coin** (probabilities)
- Example: Player A has two pure strategies $\{a, b\}$
- Her mixed strategy is the probability p to choose strategy a
- In general: if k pure strategies, a mixed strategy is any vector

$$(p_1, p_2, \ldots, p_k), \text{ s.t. } p_i \geq 0 \text{ and } p_1 + \cdots + p_k = 1$$

Example of randomization: matching pennies

		Column		
		p Heads	1-p Tails	
Row	q Heads	(1, –1)	(–1, 1)	$1p + -1(1-p) = 2p - 1$
	1-q Tails	(–1, 1)	(1, –1)	$-1p + 1(1-p) = 1 - 2p$

How to find the NE: if I randomize, I must be indifferent between the expected payoff resulting form any of the outcome

Assume an equilibrium with (q, p)

For row player to be indifferent between choosing H and T: $1p + (-1)(1-p) = (-1)p + 1(1-p)$ hence column players must use $p = 0.5$. Similarly, $q = 0.5$

Example of randomization: coordination game

coin of column player:		1/3	2/3
coin of row player:		Action Movie	Romance Movie
2/3	Action Movie	(2,1)	(0,0)
1/3	Romance Movie	(0,0)	(1,2)

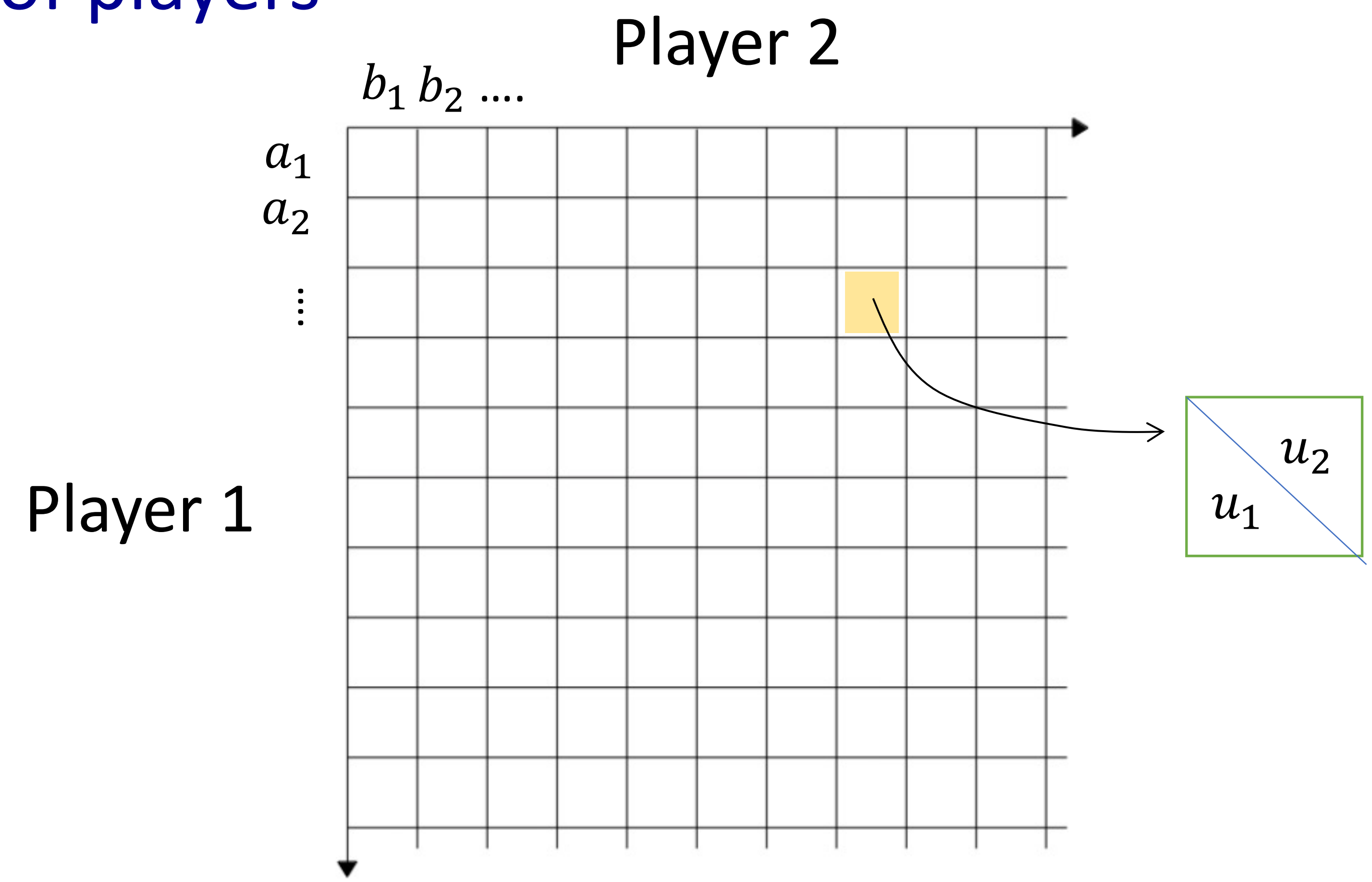
3 Nash equlibria: (A, A), (R, R), $\left(\underbrace{(\frac{2}{3}, \frac{1}{3})}_{\text{row player}}, \overbrace{(\frac{1}{3}, \frac{2}{3})}^{\text{column player}} \right)$

The DPC game definition

The DPC game

- How to construct a game out of DPC for 2 players?
  - Actions (strategies) of players
  - Payoff matrix

	Cooperate	Defect
Cooperate	3, 3	0, 5
Defect	5, 0	1, 1

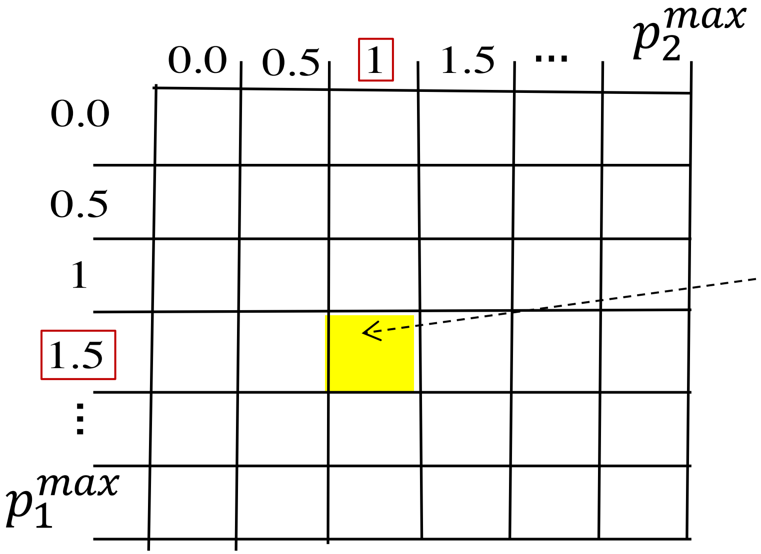


Case of 2 stations

- Discretize the power levels (to get a *finite* game)

$$u_i(p_i, p_j) = \begin{cases} -\infty & \text{if } SIR_i(p_i, p_j) < \gamma_i \\ -p_i & \text{if } SIR_i(p_i, p_j) \geq \gamma_i \end{cases}$$

- Game matrix

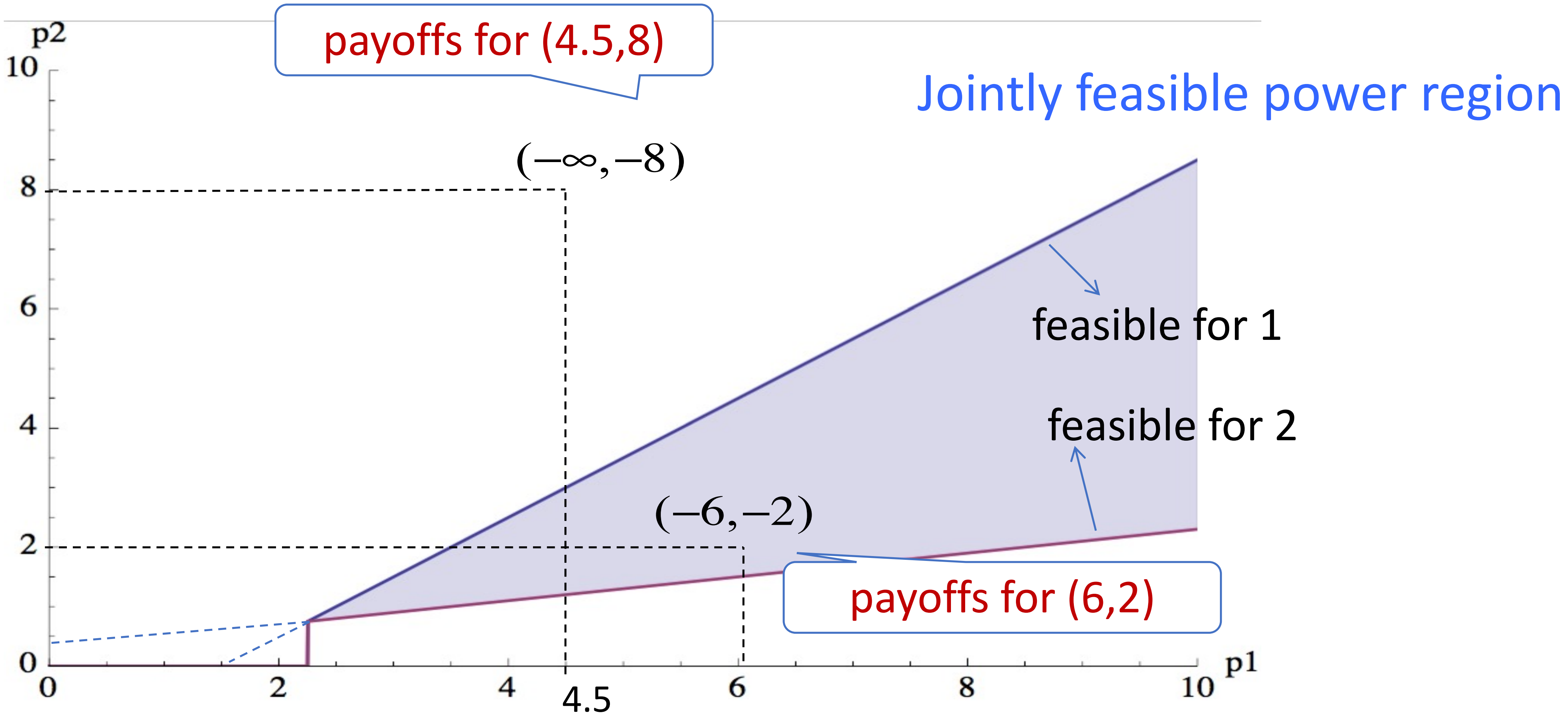


Payoffs: $u_1(1.5,1), u_2(1.5,1)$

Best-response strategies?

Nash equilibrium?

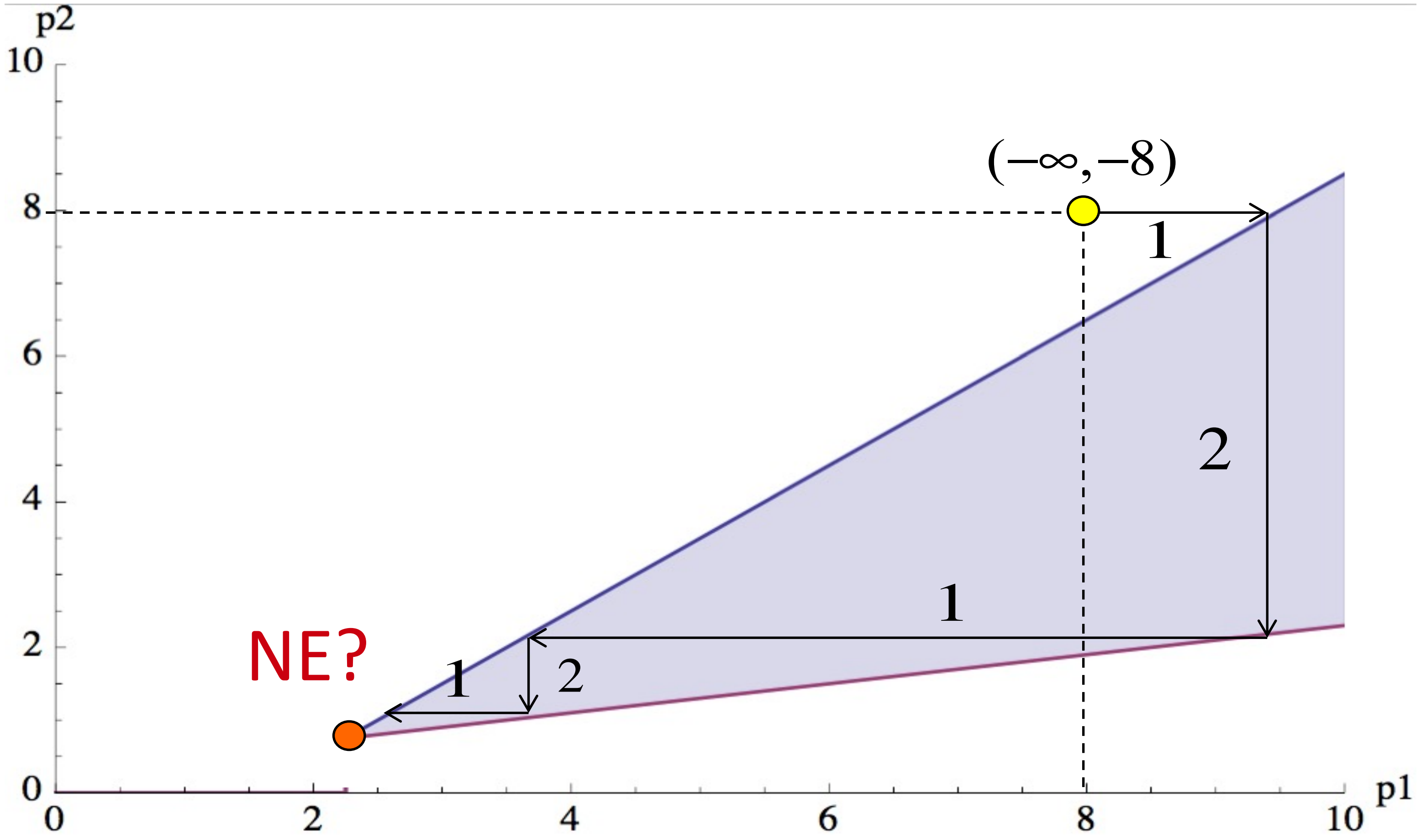
Game payoffs: based on the feasible power region



Best response strategies

What will happen if we start at (8,8) and the players alternate in their play?

Which point is an NE?



Remarks

- Best response in the previous game is the DPC algorithm!
- We observed that it converges to the **unique** NE
- NE also Social Optimum: solves the minimum total power problem! Check it in previous slide.
- Can we prove that in general? see next!

Minimize total cost
Efficiency of DPC

DPC finds optimal power vector?

- Distributed power control:

$$p_{DPC}[t+1] = D(\gamma)F\, p_{DPC}[t] + v$$

- We proved that $p_{DPC}[\infty] = (I - D(\gamma)F)^{-1}v$

iff $\rho(DF) < 1$

- Centralized optimal power selection problem:

$$p^* \triangleq \min 1^T p \quad s.t.\ (I - D(\gamma)F)p \geq v$$

- Need to prove that $p^* = p_{DPC}[\infty]$

Steps of proof

Assume feasibility, $v > 0$

feasibility and $v > 0$ implies $(I - DF)$ is an M-matrix $\Leftrightarrow$ $\rho(DF) < 1$; see https://en.wikipedia.org/wiki/M-matrix

$$p^* \triangleq \min 1^T p \quad s.t.\ (I - DF)p \geq v$$

$$= \min 1^T p \quad s.t.\ (I - DF)p = v$$

If some row i has strict inequality, it is equivalent with $SIR_i > \gamma_i$; we can then strictly reduce p_i (and hence $p_1 + \cdots + p_n$) and keep staying feasible

$$= (I - DF)^{-1} v$$

$$= \left(\sum_{k=0}^{\infty} (DF)^k\right) v$$

since $\rho(DF) < 1$

$$= p_{DPC}[\infty]$$

Corollary

For $\rho(A) < 1$, the solution of a linear system of equations of the form

$$(I - A)x = v$$

i.e., $x = (I - A)^{-1}v$, can be achieved by iterations

$$x(k+1) = Ax(k) + v$$

This is an example of the "power method": solving a system of equations by computing powers of a matrix (in our case $A^k, k = 0,1,\ldots$)

Example for $n = 2$: $(I - DF)p \geq v$

$$\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} \gamma_1 & 0 \\ 0 & \gamma_2 \end{bmatrix} \begin{bmatrix} 0 & \frac{G_{12}}{G_{11}} \\ \frac{G_{21}}{G_{22}} & 0 \end{bmatrix} \right) \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \geq \begin{bmatrix} \frac{\gamma_1 n_1}{G_{11}} \\ \frac{\gamma_2 n_2}{G_{22}} \end{bmatrix}$$

Note that each row can rewritten as $SIR_i \geq \gamma_i$

$$DF = \begin{bmatrix} 0 & \gamma_1 \frac{G_{12}}{G_{11}} \\ \gamma_2 \frac{G_{21}}{G_{22}} & 0 \end{bmatrix}$$

$$\det(\lambda I - DF) = \lambda^2 - \gamma_1 \gamma_2 \frac{G_{12} G_{21}}{G_{11} G_{22}}$$

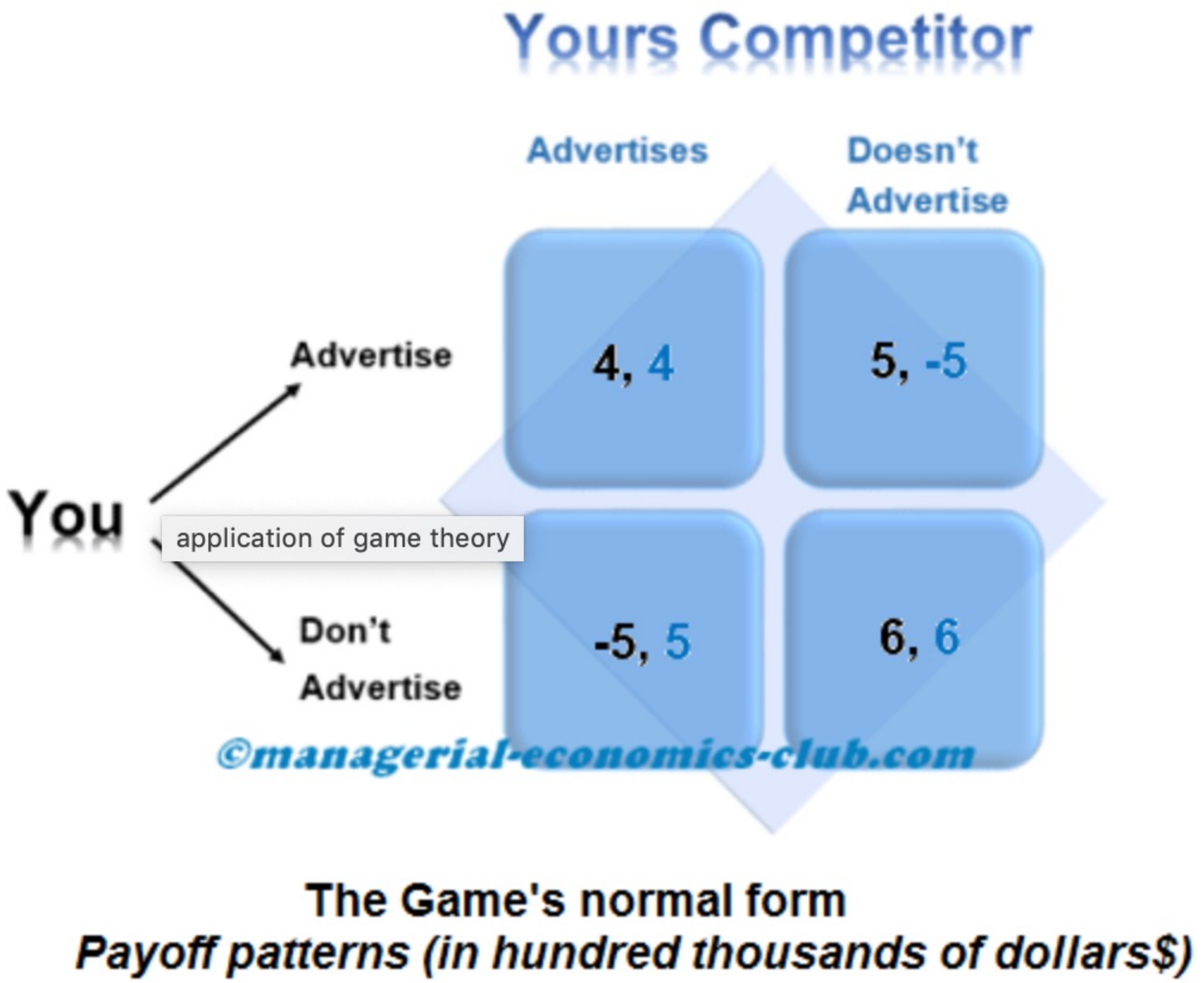
$$\rho(DF) = \sqrt{\gamma_1 \gamma_2 \frac{G_{12} G_{21}}{G_{11} G_{22}}}$$

When is $\rho(DF) > 1$?

Conclusions

- The case of CDMA is an example of an economic system where individuals gain by hurting others
- The social optimum solution of the power allocation is also the Nash equilibrium of the game that models the selfish behaviour
- DPC is a distributed algorithm that corresponds to some type of selfish behaviour that converges to the SO solution if the problem is feasible
- It is a rare case where the optimum centralized solution obtained by a central planner with full information of the system parameters is the same as the equilibrium of the distributed system where the interacting agents are selfish

Additional slides: game theory examples



Manager *(Player A)*	**Workers** *(Player B)*	
	Inefficient	**Efficient**
Does not monitor	-2, +8	+8, +4
Monitors	-8 , 0	+4, +4

Free ridding

Student "Joe"

Student "Sally"	Shirk	Work Hard
Shirk	Both recieve C's; neither works very hard on this assignment	Both recieve B's, Joe works hard but Sally does not
Work Hard	Both recieve B's, Sally works hard but Joe does not	Both recieve A's; both work very hard on this assignment

A=5, B=2, C=0, e=4

	Shirk	Work hard
Shirk	C-0, C-0	B, B-e
Work hard	B-e, B	A-e, A-e

	Shirk	Work hard
Shirk	0, 0	2, -2
Work hard	-2, 2	1, 1

Additional slides: CDMA

https://en.wikipedia.org/wiki/Code-division_multiple_access

Example of orthogonal codes

Orthogonal: $C_i \; OR \; C_j = 0$

$OR =$ inner product of the bit vectors (here we is OR instead of XOR)

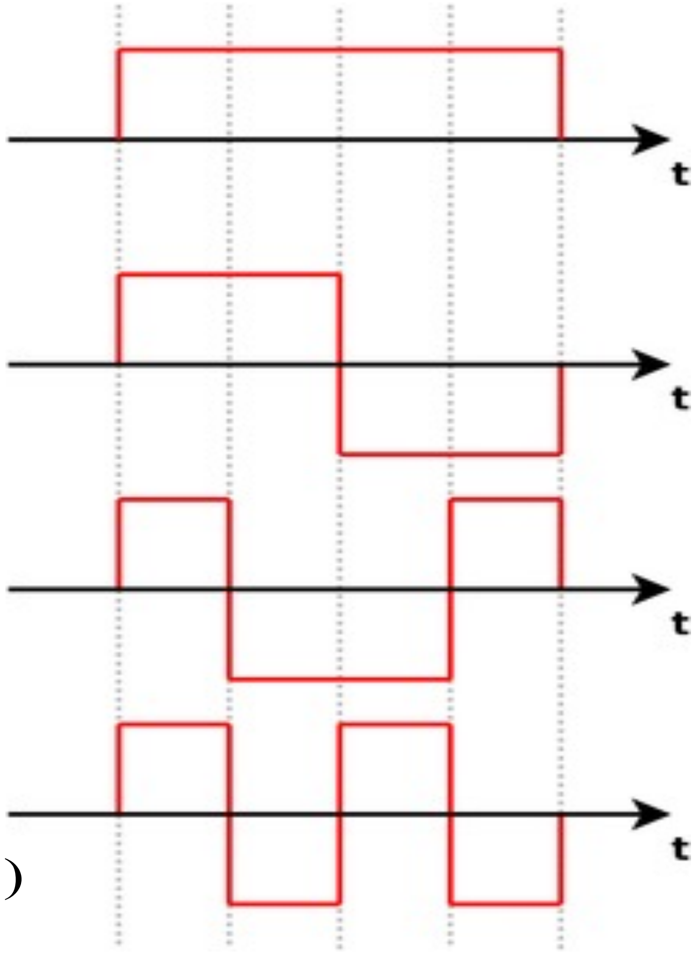
C_1

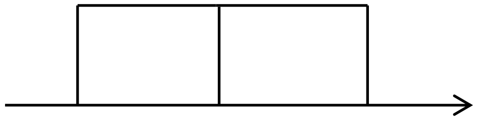
C_2

C_3

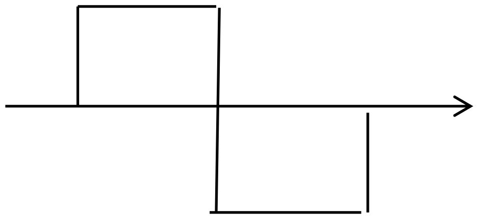
C_4

$(1,-1,1,-1)$





$(1,1)$



$(1,-1)$

Case1: Both senders send

Step	Encode sender0	Encode sender1
0	code0 = (1, –1), data0 = (1, 0, 1, 1)	code1 = (1, 1), data1 = (0, 0, 1, 1)
1	encode0 = 2(1, 0, 1, 1) – (1, 1, 1, 1) = (1, –1, 1, 1)	encode1 = 2(0, 0, 1, 1) – (1, 1, 1, 1) = (–1, –1, 1, 1)
2	signal0 = encode0 ⊗ code0 = (1, –1, 1, 1) ⊗ (1, –1) = (1, –1, –1, 1, 1, –1, 1, –1)	signal1 = encode1 ⊗ code1 = (–1, –1, 1, 1) ⊗ (1, 1) = (–1, –1, –1, –1, 1, 1, 1, 1)

Signal = signal0 + signal1 = (0, -2, -2, 0, 2, 0, 2, 0)

Step	Decode sender0	Decode sender1
0	code0 = (1, –1), signal = (0, –2, –2, 0, 2, 0, 2, 0)	code1 = (1, 1), signal = (0, –2, –2, 0, 2, 0, 2, 0)
1	decode0 = pattern.vector0	decode1 = pattern.vector1
2	decode0 = ((0, –2), (–2, 0), (2, 0), (2, 0)).(1, –1)	decode1 = ((0, –2), (–2, 0), (2, 0), (2, 0)).(1, 1)
3	decode0 = ((0 + 2), (–2 + 0), (2 + 0), (2 + 0))	decode1 = ((0 – 2), (–2 + 0), (2 + 0), (2 + 0))
4	data0=(2, –2, 2, 2), meaning (1, 0, 1, 1)	data1=(–2, –2, 2, 2), meaning (0, 0, 1, 1)

Case2: only sender0 sends

Step	Encode sender0
0	code0 = (1, −1), data0 = (1, 0, 1, 1)
1	encode0 = 2(1, 0, 1, 1) − (1, 1, 1, 1) = (1, −1, 1, 1)
2	signal0 = encode0 ⊗ code0 = (1, −1, 1, 1) ⊗ (1, −1) = (1, −1, −1, 1, 1, −1, 1, −1)

Signal = signal0

Step	Decode sender0	Decode sender1
0	code0 = (1, −1), signal = (1, −1, −1, 1, 1, −1, 1, −1)	code1 = (1, 1), signal = (1, −1, −1, 1, 1, −1, 1, −1)
1	decode0 = pattern.vector0	decode1 = pattern.vector1
2	decode0 = ((1, −1), (−1, 1), (1, −1), (1, −1)).(1, −1)	decode1 = ((1, −1), (−1, 1), (1, −1), (1, −1)).(1, 1)
3	decode0 = ((1 + 1), (−1 − 1),(1 + 1), (1 + 1))	decode1 = ((1 − 1), (−1 + 1),(1 − 1), (1 − 1))
4	data0 = (2, −2, 2, 2), meaning (1, 0, 1, 1)	data1 = (0, 0, 0, 0), meaning no data

Summary of CDMA

- Same time and same frequency resources
  - Spreading codes to differentiate
  - Non-orthogonal (“nearly-orthogonal”) allocation
- => Interference by other transmissions

- Need for interference management
  - Near-far problem
  - Feedback as solution
  - Power control: achieve target SIR for each mobile

Example: the hawk – dove game

	Player 2		
Player 1		Compromise	Don't compromise
	Compromise	(0,0)	(-1,1)
	Don't compromise	(1,-1)	(-10,-10)

Find the Nash Equilibria in pure strategies