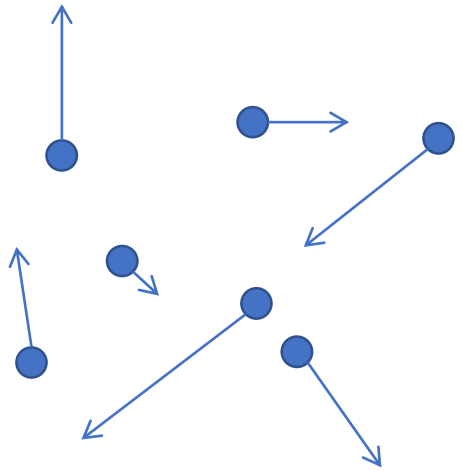
Data Structures
Programming Project #3

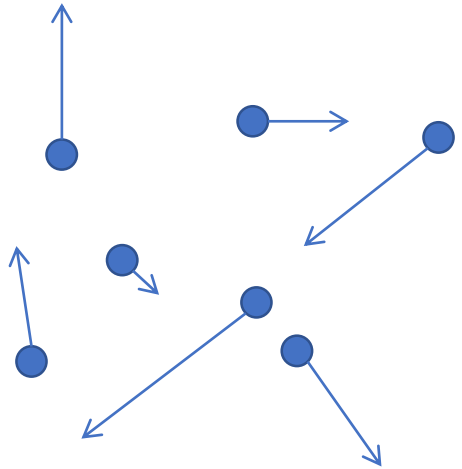
Background

- Links in wireless networks
- Want to transmit the data in the same time slot
- Impossible ∵ the interference between links
- Maximize the #links in one time slot



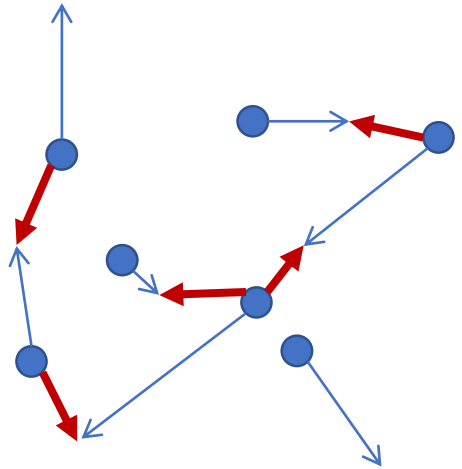
Background

- How to model the interference between links
- Option 1. Use interference graph
- Two nodes can inference each other if there is an edge between them



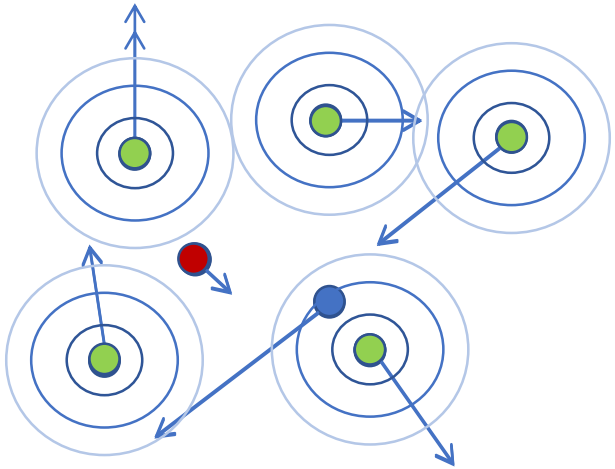
Background

- How to model the interference between links
- Option 1. Use interference graph
- Two nodes can inference each other if there is an edge between them



Background

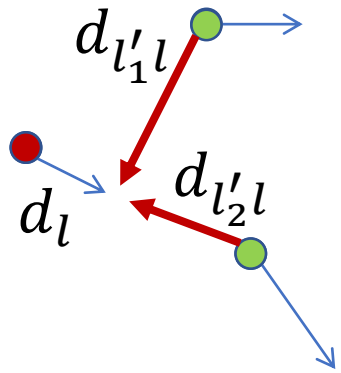
- How to model the interference between links
- Option 1. Use interference graph (inaccurate)
- Drawback: a node can be interfered with by many **far-away** nodes altogether



Background

- How to model the interference between links
- Option 2. Use SINR

$$\text{SINR}: \frac{\frac{P}{d_l{}^3}}{\sum_{other\ link\ l'\ transmitted\ with\ l} \frac{P}{d_{l'\ l}{}^3} + N} > 1$$

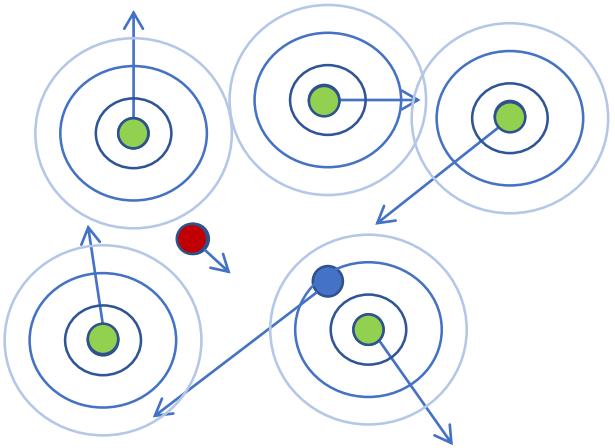


SINR of the red one: $\frac{\frac{P}{d_l{}^3}}{\frac{P}{d_{l_1'l}{}^3} + \frac{P}{d_{l_2'l}{}^3} + N}$

Background

- How to model the interference between links
- Option 2. Use SINR

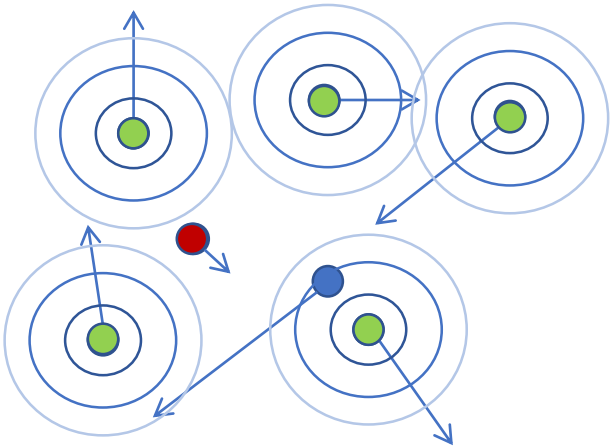
$$\text{SINR}: \frac{\frac{P}{{d_l}^3}}{\sum_{other\ link\ l'\ transmitted\ with\ l} \frac{P}{{d_{l'\ l}}^3} + N} > 1$$



Background

- How to model the interference between links
- Option 2. Use SINR (better)

$$\text{SINR}: \frac{\frac{P}{d_l{}^3}}{\sum_{other\ link\ l'\ transmitted\ with\ l} \frac{P}{d_{l'\ l}{}^3} + N} > 1$$



Background

- How to model the interference between links
- Option 2. Use SINR (better)

$$\text{SINR}: \frac{\frac{P}{d_l{}^3}}{\sum_{other\ link\ l'\ transmitted\ with\ l} \frac{P}{d_{l'\ l}{}^3} + N} > 1$$

$$\Rightarrow \boxed{\sum_{other\ link\ l'\ transmitted\ with\ l} \frac{d_l{}^3}{d_{l'\ l}{}^3} + \frac{N d_l{}^3}{P}} < 1$$

Good News and Bad News

- We choose the option 2
- **HW 3: 12% → 20%**
- Make up for your midterm exam scores

- However, the problem is NP-hard
- The optimal solution (X)
- A near-optimal solution (O)

A naïve solution

for (i = 0; i < n; i ++)

select link i if all of selected links have SINR > 1

- Try your best to find a much better one

$$\text{SINR}: \frac{\frac{P}{d_l{}^3}}{\sum_{other\ link\ l'\ transmitted\ with\ l} \frac{P}{d_{l'\ l}{}^3} + N} > 1$$

Programming Project #3: Select the wireless links

- Input:
  - All links between transmitters and receivers
  - Positions of transmitters and receivers
  - Power P and basic noise N
- Procedure:
  - Determine a set of links to transmit in the same slot
- Output:
  - The # selected links
  - The links between transmitters and receivers
- The grade is proportional to the # selected links (i.e., competition)

The Competition

- The grade is proportional to the # selected links
- Basic: 75 (deadline)
  - A feasible solution (no pair with SINR ≤ 1)
  - The # selected links is at least that by greedy algorithm
- Performance ranking (decided after the deadline)
  - [0%, 50%) (bottom): +0
  - [50%, 75%): + 5
  - [75%, 90%): + 9
  - [90%, 95%): + 12
  - [95%, 100%] (top): + 15
- Homework assistant (superb deadline)
  - +10

The Competition

- The grade is proportional to the # selected links
- Basic: 75 (deadline)
  - A feasible solution (no pair with SINR ≤ 1
  - The # selected links is at least that by gre
- Performance ranking (decided after the deadline)
  - [0%, 50%) (bottom): +0
  - [50%, 75%): + 5
  - [75%, 90%): + 9
  - [90%, 95%): + 12
  - [95%, 100%] (top): + 15
- Homework assistant (superb deadline)
  - +10

Input Sample: use scanf

Format:

#Nodes #Links Power Baisc_Noise
Node_ID X_Pos Y_Pos
…
Link_ID Link_End1 Link_End2
…

Output Sample: use printf

Format:

#AcceptedLinks
Link_ID Link_End1 Link_End2
…

Note

- Superb deadline: 12/14 Tue
- Deadline: 12/21 Tue
- Pass the test of our online judge platform
- Submit your code to E-course2
- Demonstrate your code remotely with TA
- **C Source code (i.e., only .c)**
- Show a good programming style