ICPC 2023 Online Fall Challenge powered by Huawei

Ideas by Gleb Astashkin

(aka gleb.astashkin)

A few words about myself

- It’s my third cash prize in the Huawei ICPC completions
- I finished 7^{th}, it’s my personal best
- I used to work at telecommunication company, that’s why I know something about transmitting signals from base stations to client’s devices
- It didn’t help me at all during the competition

Speech plan

- General approach – greedy solution to beat most of the cases
- Packing multiple frames to one (cell, radio)
- Unique approach to deal with low duration frames + low average D cases
- Impressions and additional thoughts

General approach

General approach

- There is no penalty if an user occupies a radio exclusively

$$s_{rnt}^{(k)} = \frac{s_{0,rnt}^{(k)} \times p_{rnt}^{(k)} \times \prod_{m \neq n} e^{d_{mrn}^{(k)} \times b_{rmt}^{(k)}}}{1 + \sum_{k' \neq k, n' \neq n} s_{0,rnt}^{(k')} \times p_{rn't}^{(k')} \times e^{-d_{n'rn}^{(k')}}}$$

$$s_{rnt}^{(k)} = s_{0,rnt}^{(k)} \times p_{rnt}^{(k)}$$

- Let’s sort all the frames somehow, iterate through them and try to schedule them one by one greedy
- Since radio resources are limited, we will try to minimize consumption of radio resources
- And use power consumption as a tiebreaker

General approach – frame ordering

- Since this is a heuristic, different way to order the frames leads to different results
- Make multiple runs using different ordering and choose the best
- We can order frames by:
  - Occupied radios
  - Occupied pairs (cell, radio)
  - Purely random
  - Combination of all above

General approach – extra tricks

- Occupy only limited amount of power on the first run – set power limit for (cell, radio) to 1.0 instead of min(4.0, R).

  Utilize all the energy on the last run

- Recalculate the order of frames based on already occupied resources, for example between the first and the last runs

General approach – multiple runs

Consider the case where:

- K = 4
- R = 5
- First two cells have twice greater S_0 then, the others
- There are 4 frames with total ‘size’ = 6
- A frame occupies radio exclusively: 1 radio = 1 frame

S_0	K_0	K_1	K_2	K_3
R_0	2	2	1	1
R_1	2	2	1	1
R_2	2	2	1	1
R_3	2	2	1	1
R_4	2	2	1	1

General approach – multiple runs

Without power limit

P	K_0	K_1	K_2	K_3	Sum
R_0	3	0	0	0	6
R_1	2	1	0	0	6
R_2	0	3	0	0	6
R_3	0	1	4	0	6
R_4	0	0	1	4	5
Sum	5	5	5	4	

Power limit = 1.0

P	K_0	K_1	K_2	K_3	Sum
R_0	1	1	1	1	6
R_1	1	1	1	1	6
R_2	1	1	1	1	6
R_3	1	1	1	1	6
R_4	1	1	1	1	6
Sum	5	5	5	5	

Packing multiple frames to one (cell, radio)

Packing multiple frames to one (cell, radio)

- Penalty for doing that is not that big. It doesn't involve P_{rnt} at all

$$s_{rnt}^{(k)} = \frac{s_{0,rnt}^{(k)} \times p_{rnt}^{(k)} \times \prod_{m \neq n} e^{d_{mrn}^{(k)} \times b_{rmt}^{(k)}}}{1 + \sum_{k' \neq k, n' \neq n} s_{0,rnt}^{(k')} \times p_{rn't}^{(k')} \times e^{-d_{n'rn}^{(k')}}} \quad \Rightarrow \quad s_{rnt}^{(k)} = s_{0,rnt}^{(k)} \times p_{rnt}^{(k)} \times \prod_{m \neq n} e^{d_{mrn}^{(k)} \times b_{rmt}^{(k)}}$$

- Usually we have some leftover power, so we can utilize it by packing frames
- It's perfect when we have many 1 time unit duration frames
- There's no way to improve final score if average D is high enough

Packing multiple frames – algorithm

Since we share radio resources between multiple frames, we have to consider interference – frames influence to each other. Deal with it using following steps:

1. Assign all available frames to the chosen (cell, radio)
2. Use bin search to calculate minimum amount of power to complete the frame
3. Choose maximum subset of frames we can complete under given constraints
4. Add a few runner ups and repeat all previous steps
5. Stop if the subset of frames is stable

Unique approach for low duration frames and low average D cases

Unique approach

- We are not allowed to use same radio on multiple cells due to high penalty

$$s_{rnt}^{(k)} = \frac{s_{0,rnt}^{(k)} \times p_{rnt}^{(k)} \times \prod_{m \neq n} e^{d_{mrn}^{(k)} \times b_{rmt}^{(k)}}}{1 + \sum_{k' \neq k, n' \neq n} s_{0,rnt}^{(k')} \times p_{rn't}^{(k')} \times e^{-d_{n'rn}^{(k')}}}$$

- But we are allowed to use multiple radios for the same frame
- Actually we can even share set of (cell, radio) between multiple frames
- The only penalty is $\prod_{m \neq n} e^{d_{mrn}^{(k)} \times b_{rmt}^{(k)}}$

Unique approach – set of (cell, radio)

Choose the set of (cell, radio) in such way, that:

1. A radio used by only one cell
2. Maximum number of radios assigned to a single cell is as lower as possible

Then choose maximum subset of frames using algorithm from the previous part and scheduling each frame to all chosen (cell, radio) at once

P	K_0	K_1	K_2	K_3	Sum
R_0	4				4
R_1		4			4
R_2			4		4
R_3				2.5	2.5
R_4				2.5	2.5
Sum	4	4	4	5	

Unique approach – a little trick

Some of the frames are so small we can schedule them to a smaller amount of radios (perhaps 1 or 2)

Assign them only on orange (cell, radio)s

It slightly reduces penalty for all the other frames scheduled on green (cell, radios)

Make it in two runs:

1. Assign some frames only on orange (cell, radio)s
2. Try to assign maximum number of unassigned frames on all (cell, radio)s: both green and orange

P	K_0	K_1	K_2	K_3	Sum
R_0	4				4
R_1		4			4
R_2			4		4
R_3				2.5	2.5
R_4				2.5	2.5
Sum	4	4	4	5	

Impressions and additional thoughts

Impressions and additional thoughts

- All the formulas seemed scary at first, but turned out it was not that bad
- Low TL made feedback loop short and convenient. Also it made the problem deeper, since we have not enough runtime to do classical SA
- Shared testcases made local debugging possible. I like that number of them was small to prevent overfitting
- Once again I don’t think probing is a problem. At least system tests deal well with overfitted solutions
- Difference between preliminary and system tests was small. Set of top-10 competitors didn’t change after the system tests. That’s a good sign
- Overall it was a great contest! I really enjoyed it. Thank you!

Thank you for your attention!

謝謝你！

Telegram: @gvastash

Linkedin: linkedin.com/in/gvastash