Optimizing traffic scheduling with self-adaptive methods

Sixiao Zhu, Peizhen Hu

Mar. 2015

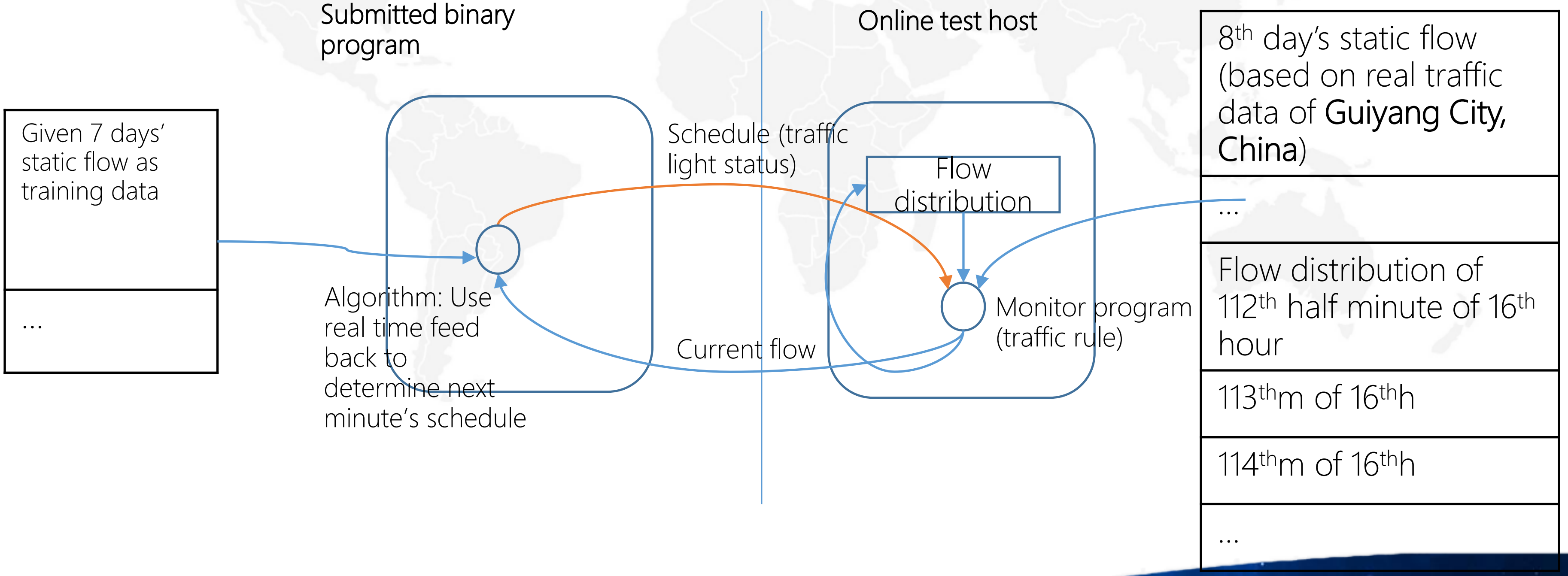
Intro

- Team name : foo

- members :
    Sixiao Zhu (朱思晓), SCUT
        Coding, debuging
    PeiZhen Hu (胡培真), SCUT
        Algorithm, parameters

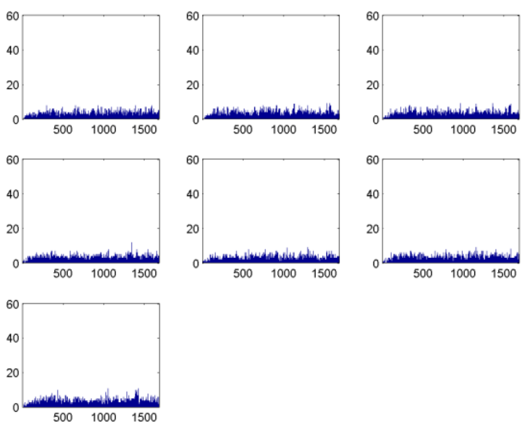
Understanding to "intellectual traffic scheduling"

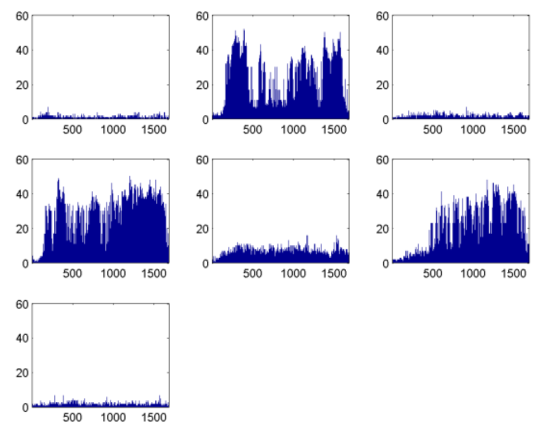
- This is more of a abstract, complex optimization problem, rather than a real traffic problem.



Data analysis: Time Domain

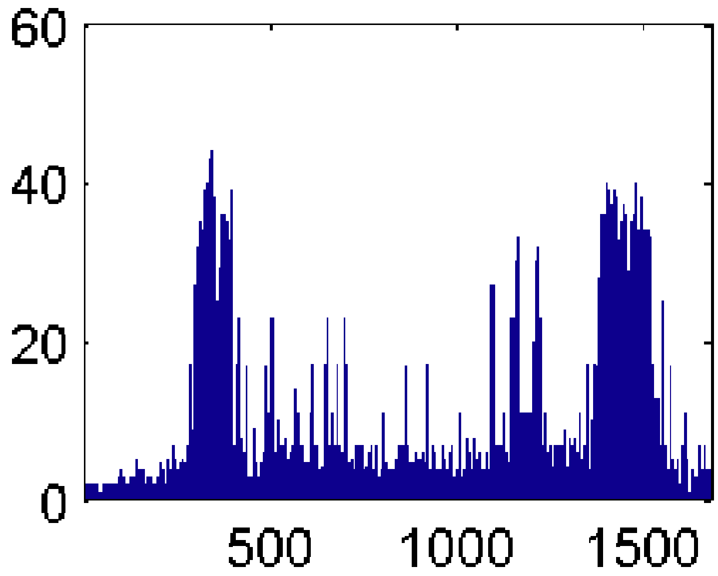
- On the granularity of “day”, some roads are stable across the given 7 days, some seem irregular.





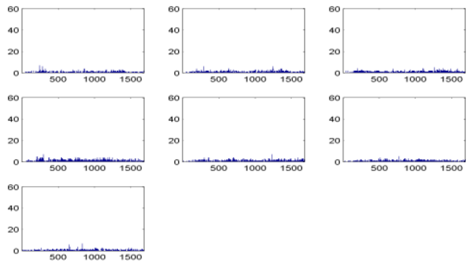
Data analysis: Time Domain

- On the granularity of “Hour”, traffic may burst.

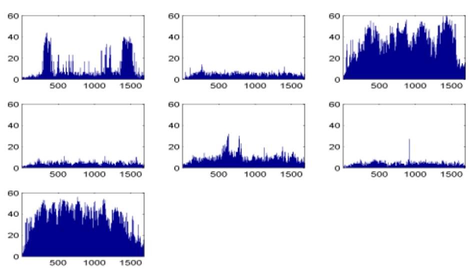


Data analysis: Time Domain

- For the traffic on the 8th day, some roads could be reasonably predicted offline with average of training data. (with certain scaling of course, we noticed that 8th day's average traffic is lower than normal), but these are mostly low-traffic roads, which is not significant.

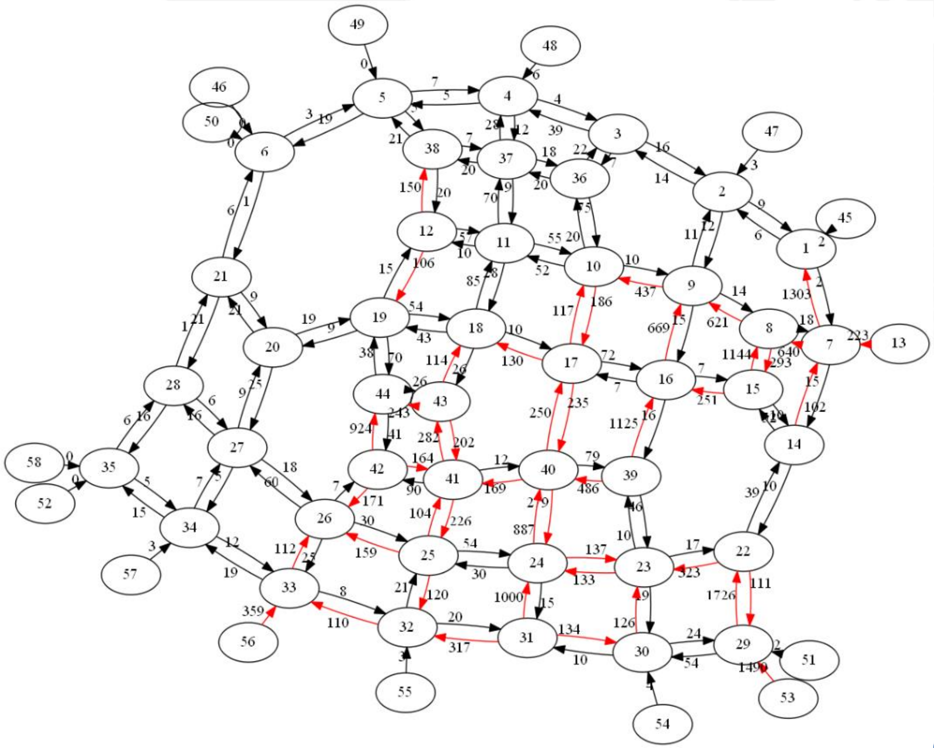


- Most busy roads behave unreasonably across the given 7 days.



Data analysis: Space Domain

- Flow distribution at the end of 9th hour. Heavily roads are marked red.



Data analysis: Space Domain

- On analyzing of the given evaluation code, we found that flow will “dispear” in the map border roads.
- Disapeared flow will not count in the future loss, thsu a obvious strategy is to make more traffic flow to this edges.

Data analysis: Space Domain

- F as current flow on the road, v as export rate, assume no flow import, we define the $loss\ of\ F$ as the sum of all
- $loss = \sum_{i=1}^{F/v} F - iv = \frac{F^2}{2v}$
- Total loss is proportional to flow's square, thus busy roads are particularly significant.
- This non-linear property makes the flow scale matters to the schedule strategy. We use scaled training data to 0.8 and got better online judge score, this indicates the 8^{th} days average flow is less than the first 7 days.

Candidate model: Greedy

- For every cross, enumerate every possible traffic light combination to make local loss optimal.
- Not global optimal.

Candidate model：Network flow

- Minimum-cost maximum flow
- Assign big cost to busy road, small cost to relaxing road, assign 0 cost to the board roads that are “exportable”.
- Use edges with flow more than out rate as source, edges with flow less than out rate and “exportable” edges as destination, use minimum-cost maximum flow algorithm to get a schaduel that minimize the cost.
- Pros: This model could balance the flow distribution and ensure the “exporting” rate.
- Cons: No way to add in traffic rule constrain.

Candidate Model : Offline training (simulated annealing + random search)

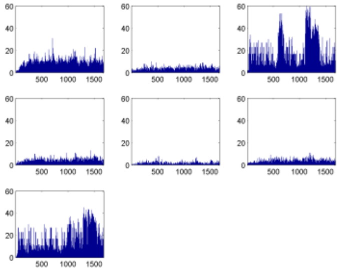
- Scale the training data of 7th day, use random search to get to best schedule for this day, and use it directly to 8th days schedule.
- Schedule for the 14 test hours are exactly same, we used this strategy to get 269w loss.
- Training method:
  1. Generate initial schedule with local greedy method.
  2. Flip a part of the lights(green turns red, or red turns green), the flip number reduces while the system gets stable.
  3. Feed the flipped schedule to monitor, and calculate the new loss.
  4. If improved, accept the change, otherwise discard the change.
  5. Go back to 2.

Model candidate : Offline training (optimize for each hour individually)

- The flow scale for each hour differs, according to the previous conclusion, this leads to different optimal schedule.
- We tried to train individually for each hour, though offline score improved significantly, online test score got worse.
- The reason for this may be over fitting, in previous method, the 14 hours' total loss is used to evaluate a flip, while in this method, only the hour in which a flip lays. The small scale feed back brings more over fit to specific training data.

Final model: Offline training + Online adjustment

- Innovation: the shortage for offline training is that we can not perfectly fit the 8th day's flow distribution.



- (in the figure, optimal schedule for the 7 days are largely different, so we do not know what will 8th day be like)
- Solution: When testing, we use the feed back flow to calculate the 8th day's flow. Use this flow to make a short term prediction and continue training our schedule.

Final model: Offline training + Online adjustment

- Detailed description:
  1. Get current minute flow from monitor.
  2. Calculate the “static flow” of current minute based on schedule previous minute. Use the last 30 minutes’ static flow obtained this way to predict the future 30 minutes.
  3. Retrain the offline obtained schedule using the predicted 30 minute flow, with the same strategy as offline training.
  4. Output the retrained schedule to monitor.

Final model: Offline training + Online adjustment

- Some tracks for speed up:
  - Use 12bit binary to represent light combination, preprocess traffic rule violation penalty under each circumstance.
  - Incremental update, calculate each flip's affect fast.

Thanks