Scheduling for moving vehicles with garantee Proportional Fairness between users

Nga NGUYEN

Supervisors: Olivier Brun, Balakrishna Prabhu

Fréjus Summer School
28 June 2018

Short Bio

HNUE
HANOI NATIONAL UNIVERSITY OF EDUCATION

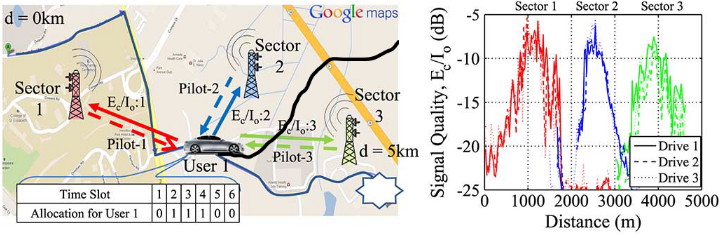
Université
Paul Sabatier
TOULOUSE III

LAAS
CNRS

Continental

Introduction

- Current scheduling algorithms use current/past information for decision.
- Connected vehicles technology will gives access to future information (path prediction+ Signal-to-noise ratio (SNR) maps $\rightarrow$ predict future rate)



(a) (b)

Can data on future improve efficiency of algorithms?

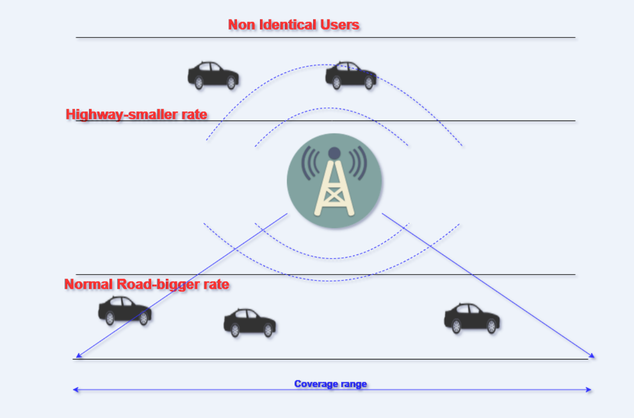


Figure: Drive-thru Internet systems.

- For each time the base station (BS) allows at most one vehicle to allocate data.

Objective: Guarantee Proportional Fair and High Throughput

$$\text{maximize } C = \sum_{i=1}^{K} log \left(\sum_{j=1}^{T} \alpha_{ij} r_{ij} \right)$$

$$\text{subject to } \sum_{i=1}^{K} \alpha_{ij} = 1, \alpha_{ij} \in \{0, 1\}$$

- K is number of cars, T is number of time slots, $\alpha_{i,j}$ is the allocation, $r_{i,j}$ is the rate.
- log objective function stands for Proportional fairness between users.

$\rightarrow$ $r_{i,j}$ is **given**, $\alpha_{i,j}$ is **variable**.

Some Existing Algorithms

- **Greedy**. Current info only: choose the vehicle with best current rate.
- **PF-EXP**[3]. Current + past info: choose user with best

$$\frac{\text{current rate}}{\text{total rate in the past}}$$

  - Used in 3G network; optimal in some cases (not necessarily true for road traffic)
- **Discrete Gradient** [2]. Use current + past + future info: choose user with the best

$$\frac{\text{current rate}}{\text{total rate in the past} + \text{current} + \text{estimated future rate}}$$

- Relax the integer constraints.

$$\text{maximize } C = \sum_{i=1}^{K} \log\left(\sum_{j=1}^{T} \alpha_{ij} r_{ij}\right)$$

$$\text{subject to } \sum_{i=1}^{K} \alpha_{ij} = 1, \alpha_{ij} \in [0, 1]$$

Projected Gradient: Upper bound- relaxed problem

- Relax the integer constraints.

$$\text{maximize } C = \sum_{i=1}^{K} \log\left(\sum_{j=1}^{T} \alpha_{ij} r_{ij}\right)$$

$$\text{subject to } \sum_{i=1}^{K} \alpha_{ij} = 1, \alpha_{ij} \in [0, 1]$$

- Benchmark algorithm for comparison

Formula for projected gradient

Update rule

- Start with $\alpha^{(0)} \in D$ where D is the feasible set.
- $\alpha^{(n+1)} = \Pi_D(\alpha^{(n)} + \epsilon_n \nabla C(\alpha^{(n)}))$, with $\epsilon_n \in (0, 1)$ is learning rate at step n.

$\rightarrow$ This update rule can **approach the global optimal**.
In this below we compute the projection $\Pi_D(\alpha^{(n)} + \epsilon_n \nabla C(\alpha^{(n)}))$.

Lemma

Given A, B two finite sets, A is nonempty set. Then there exists a decomposition $B = B^1 \sqcup B^2$ such that $\text{mean}(A \cup B^1) \leq i$ for every $i \in B^1$ and $\text{mean}(A \cup B^1) > i$ for every $i \in B^2$. With convention that boolean on empty set is always true.

Ex: $A = \{5\}, B = \{1, 4, 6\}$ then $B^1 = \{6\}$ and $B^2 = \{1, 4\}$.

Fix $\alpha^{(n)}$, denote $\nabla_{i,j}C = \partial C/\partial\alpha_{i,j}(\alpha^{(n)})$.

Proposition

$\alpha^{(n+1)}$ is given in the following formula: For each j, define
$A(j) = \{\nabla_{i,j}C | \textit{for } i \textit{ s.t } \alpha_{i,j}^{(n)} > 0\}$ *and*
$B(j) = \{\nabla_{i,j}C | \textit{for } i \textit{ s.t } \alpha_{i,j}^{(n)} = 0\}$.
$B^1(j), B^2(j)$ *are defined as lemma 4.1 for two set* $A(j), B(j)$ *and*
$\bar{m} := \mathrm{mean}(A(j) \cap B^1(j))$.
Then

$$\alpha_{i,j}^{(n+1)} = \begin{cases} \alpha_{i,j}^{(n)} \textit{ if } \alpha_{i,j}^{(n)} = 0 \textit{ and } \nabla_{i,j}C \in B^2(j) \\ \alpha_{i,j}^{(n)} + \epsilon_n(\nabla_{i,j}C - \bar{m}) \textit{ otherwise }. \end{cases}$$

Proposition

If $\alpha^* \in D$ and $\tilde{\nabla} C(\alpha^*) = 0$ then α^* is the optimal value of the relax problem, where

$$\tilde{\nabla} C(\alpha^*) = \begin{cases} 0 \textit{ if } \alpha^*_{i,j} = 0 \textit{ and } \nabla_{i,j} C(\alpha^*) \in B^2(j) \\ \nabla_{i,j} C(\alpha^*) - \bar{m} \textit{ otherwise} \end{cases}$$

With this notation, $\alpha^{(n+1)} = \alpha^{(n)} + \epsilon_n \tilde{\nabla} C(\alpha^{(n)})$. So if we do algorithm until $\tilde{\nabla} C(\alpha^{(n)})$ coverges to 0, it implies $\alpha^{(n)}$ converges to α^* at which $\tilde{\nabla} C(\alpha^*) = 0$, i.e, α^* is the optimal.
$\rightarrow$ We do recursion until $\tilde{\nabla} C(\alpha^{(n)})$ coverges to 0.

Simulations

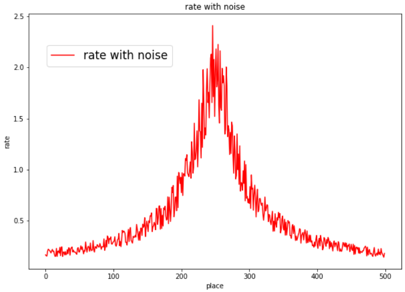


Figure: Rate in coverage range of one Base Station

Moving vehicles and Measurement based

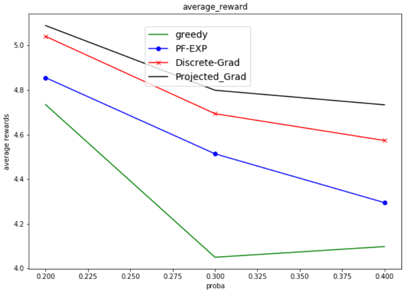


Figure: Rate in coverage range of one Base Station

Michael J. Neely,
Exploiting Mobility in Proportional Fair Cellular Scheduling: Measurements and Algorithms,
IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 24, NO. 1, FEBRUARY 2016.

Robert Margolies et Al,
Energy Optimal Control for Time Varying Wireless Networks,
IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 52, NO. 7, JULY 2006.

H. J. Kushner and P. A. Whiting,
Asymptotic Properties of Proportional-Fair Sharing Algorithms: Extensions of the Algorithm,
IEEE Transactions on Wireless Communications, VOL. 3, NO. 4, JULY 2004

Thank you for your attention!
Any question?