Balancing a bike-sharing system in Real Time

Daniel Chemla, Frédéric Meunier, Roberto Wolfler-Calvo,
Houssame Yahiaoui, Thomas Pradeau

June 2011

GT-LG, Paris

Talk plan

- Introduction and motivation
- Modelization
- Exploitation methods
  - Using a truck
  - Online Tarification
- Evaluation
  - Simulator presentation
  - Results

Motivation: improving regulation issue

- *It is statistically impossible to be sure to find a bike or a park place in 100% of the cases*
  Albert Asseraf, Strategy and Marketing France Chief Executive, JCDecaux
  Le Figaro, 26 mars 2010

Motivation: improving regulation issue

- *It is statistically impossible to be sure to find a bike or a park place in 100% of the cases*
  Albert Asseraf, Strategy and Marketing France Chief Executive, JCDecaux
  Le Figaro, 26 mars 2010

System modelization

- Clients arrival rate at a station: Poisson process
- Destination choice: O-D matrix generated with gravity model
- Goal : improving client satisfaction
  - No bike: the client leaves the system unsatisfied
  - No paking: the client leaves the system unsatisfied and with the bike
- $\mathbf{S}$ the set of stations, $|\mathbf{S}| = \mathbf{n}$ and $\mathbf{m_n}$ the total number of bikes

The Queue Modelization

- A station is modelled as a M/M/1 queue where servers are the users and bikes are in the queues (infinite capacity)
- The time spent in trajectory is modelled as a M/M/∞ queue
- $$\mathbf{N_n(t) = \{N_{ij}(t), (i, j) \in S^2, t \geq 0\}}$$

  such that:
  - $\mathbf{N_{ij}(t)}$ number of bikes going from station $\mathbf{i}$ to $\mathbf{j}$ at time $\mathbf{t}$
  - $\mathbf{N_{ii}(t)}$ the number of bikes parked are station $\mathbf{i}$

The former process $\mathbf{N_n(t), t \geq 0}$ is an irreductible Markov Chain and has an invariant probability that is not easy to compute

The Queue Modelization

Asymptotic approximation to have exploitable results:

- Open network of independant network of $\mathbf{N}^2$ queues
- The probabilty of the open network has a product form
- Asymptoticly the original network in its stationnary state is behaving as the open network

Exploitation system

- Improving efficiency: avoid stations to be empty or full
- First way: ordering trucks to balance the system, moving bikes from attractive stations to repulsive ones
- Second way: without trucks, incitating people to regulate the system by encouraging them to park bikes in empty stations

Exploitation system

Heuristic using trucks need a target state defined for each station.
Heuristic here are made for one truck

First heuristic: objective driven

- The truck is sent to the two most unbalanced stations
- Arriving at the station it tries to balance it for the best
- A new call to the operating system every two moves
- The evolution of the system not taken into account

Exploitation system

Heuristic using trucks need a target state defined for each station.
Heuristic here are made for one truck

First heuristic (bis): objective driven

- The truck is sent to the two most unbalanced stations - with a correction using the incoming flows of bikes
- Arriving at the station it tries to balance it for the best
- A new call to the operating system every two moves
- The evolution of the system not taken into account

Exploitation system

Second heuristic: DP

The truck is defined by its **state**. At timestep $\mathbf{k}$

$$\mathbf{E_k = (V_k, p_k, l_k, t_k) \in S_n^k \times V_k \times [|0, K_c|] \times [|0, T_{max}|]}$$

where

- $\mathbf{V_k}$: list of the already visited stations since the start of the mission $|\mathbf{V_k}| = \mathbf{k}$,
- $\mathbf{p_k}$: truck position
- $\mathbf{l_k}$: truck load. ($\mathbf{K_c}$ = truck capacity)
- $\mathbf{t_k}$: time slop elapsed since the start of the mission

For each station $\mathbf{i} \in \mathbf{S}$ a target state $\mathbf{T_i}$ is defined

Exploitation system

Cost of going from state $\mathbf{E_k}$ to station $\mathbf{E_{k+1}}$:

$$\mathbf{J(E_k, E_{k+1}) = |T_i - (N_{ii}(\overline{t_i}) + L(I_k, N_{ii}(\overline{t_i}))| + \sum_{j \notin \{V_k \cup \{i\}\}} |T_j - N_{jj}(\overline{t_i})|}$$

where $\mathbf{p_{k+1} = i \notin V_k}$
$\mathbf{\overline{t_i} = t_k + T_{V_k,i}}$

Exploitation system

Cost of going from state $\mathbf{E_k}$ to station $\mathbf{E_{k+1}}$:

$$\mathbf{J(E_k, E_{k+1}) = |T_i - (N_{ii}(\overline{t_i}) + L(I_k, N_{ii}(\overline{t_i}))| + \sum_{j \notin \{V_k \cup \{i\}\}} |T_j - N_{jj}(\overline{t_i})|}$$

where $\mathbf{p_{k+1} = i \notin V_k}$
$\mathbf{\overline{t_i} = t_k + T_{V_k,i}}$
▷ Backward DP thanks to Bellman equation to obtain the best command for the truck

$$\mathbf{J^*(E_k) = \min_{s \notin V_k} \mathbb{E}\left[J(E_k, s) + J^*(E_{k+1}(s)) | N_s(t_k)\right]}$$

In practice, experiments done for small cities (up to **20** stations) and **2** to **3** timesteps

Exploitation system

Third heuristic: The Colored Cluster balancing approach

- The truck's optimal decision is done taken into account the number of bikes at each stations and on the trucks
- a huge number of states if taken all these informations into account
- Clustering stations: a client does not mind changing stations if they are a few tens of meters away
- Coloring Cluster: the optimal decision should not be very depending of the exact number of bikes in a cluster but on the average level of filling; 3 levels are defined: deficit of bikes, average filling, excess of bikes
- $\longrightarrow$ The number of states is then $\mathbf{NbStates} = \mathbf{3}^{\mathbf{NbCluster}} * \mathbf{3} * \mathbf{NbCluster} = \mathbf{13122}$ for **6** clusters.

Exploitation system

- With the probability matrix to go from a state to another and defining a target state in which all stations are balanced we can find the optimal policy to get to the target state for the least mean cost
- The optimal policy is obtained thanks to a classical policy iteration algorithm
- Problem : obtain the probability matrix
  - With the Queue modelization
  - With a nanosimulator

Exploitation system

Fourth heuristic: The Online Tarification Approach

- Objective: Regulate **without any truck**
- Control: Prices on arrival stations
- $\longrightarrow$ Defining a targeted level of filling for all stations **TargetFilling**
- $\mathbf{x^k_{(i,j)}} \geq \mathbf{0}$: People that wanted to go from station **i** to **j** but park at station **k** instead
- $\mathbf{c^k_{(i,j)}} = \mathbf{C^B_{(i,k)}} + \mathbf{C^F_{(k,j)}} - \mathbf{C^V_{(i,j)}}$: Cost to stop at station **k** instead of **j** and walk to station **j**

Exploitation system

$$
\begin{aligned}
\text{Min} \quad & \sum_{(i,j,k)\in S^3} x^k_{(i,j)} c^k_{(i,j)} & & & \\
\text{s.t.} \quad & \sum_{(i,j)\in S^2} x^k_{(i,j)} &= \quad & T_k & \text{for all } k \in S & \quad \text{(i)} \\
& \sum_{k\in S} x^k_{(i,j)} &= \quad & \gamma\lambda_i P_{ij} & \text{for all } (i,j) \in S^2 & \quad \text{(ii)} \\
& x^k_{(i,j)} \geq 0 & & & \text{for all } (i,j,k) \in S^3 & \quad \text{(iii)}
\end{aligned}
\tag{1}
$$

Where

- $T_k = \max\{0, TargetFilling - Load_k\}$: current default in bikes
- λ_i: Mean arrival rate per station
- $\gamma = \frac{\sum_{k\in S} T_k}{\sum_{k\in S} \lambda_k}$: normalization constant

Exploitation system

$$
\begin{array}{llcll}
\text{Max} & \displaystyle\sum_{(i,j)\in S^2} \gamma\lambda_i P_{ij}\beta_{(i,j)} + \sum_{k\in S} T_k\mu_k & & & \\
\text{s.t.} & c^k_{(i,j)} - \mu_k - \beta_{(i,j)} & \geq & 0 & \text{for all } (i,j,k)\in S^3 \quad (i) \\
& \beta_{(i,j)}, \mu_k \in \mathbb{R} & & & \text{for all } (i,j,k)\in S^3 \quad (ii)
\end{array}
\tag{2}
$$

Exploitation system

$$
\begin{array}{llll}
\text{Max} & \sum_{(i,j)\in S^2} \gamma \lambda_i P_{ij} \beta_{(i,j)} + \sum_{k\in S} T_k \mu_k & & & \\
\text{s.t.} & c^k_{(i,j)} - \mu_k - \beta_{(i,j)} & \geq & 0 & \text{for all } (i,j,k) \in S^3 & \text{(i)} \\
& \beta_{(i,j)}, \mu_k \in \mathbb{R} & & & \text{for all } (i,j,k) \in S^3 & \text{(ii)}
\end{array} \quad (2)
$$

▷ a set of dual prices $\{\mu_k, k \in S\}$

Exploitation system

$$
\begin{array}{llcll}
\text{Max} & \sum_{(i,j)\in S^2} \gamma\lambda_i P_{ij}\beta_{(i,j)} + \sum_{k\in S} T_k\mu_k & & & \\
\text{s.t.} & c^k_{(i,j)} - \mu_k - \beta_{(i,j)} & \geq & 0 & \text{for all } (i,j,k)\in S^3 \quad \text{(i)} \\
& \beta_{(i,j)}, \mu_k \in \mathbb{R} & & & \text{for all } (i,j,k)\in S^3 \quad \text{(ii)}
\end{array} \tag{2}
$$

▷ a set of dual prices $\{\mu_k, k \in S\}$

▷ when a client appears at station $\mathbf{i}$ and want to go to $\mathbf{j}$ he can go to $\mathbf{k}$ instead for the following cost:

$$u(k) = c^k_{(i,j)} - \mu_k$$

Each client chooses the solution that has the least cost for him.

$$\beta^*_{(i,j)} = \min_{k\in S} c^k_{(i,j)} - \mu_k$$

is the price that will finally pay a client who wants to go from $\mathbf{i}$ to $\mathbf{j}$. He will go to station $k^* \in S$ such that $c^{k^*}_{(i,j)} - \mu_{k^*} = \beta^*_{(i,j)}$

Evaluation of the system

Simulator

- Clients are generated with respect to a Poisson process
- Their targeted destination is taken with respect to a O-D matrix that has been generated with a gravity model
- The time elapsed while driving from a station to another is computed with respect to the distance and altitude between two stations
- Clients who do not find bikes or parking spots can visit several stations before leaving the system with respect to a bound in time and stations given by their profil type

Indicator

- Number of satisfied clients
- Number of clients who did not find a bike
- Number of clients who did not find a parking
- Number of clients who change their targeted station for another one (tarification approach)

Evaluation of the system

Result: Low case demand - edoras

Size	Indicator	Empty	OB	OB-corr	DP	CC	OT
20	Satisfied	**677**	**962**	**963**	**738**	**714**	**829 + 156**
	No bikes	**199**	**1**	**1**	**154**	**171**	
	No parking	**124**	**29**	**28**	**104**	**124**	
50	Satisfied	**2027**	**2265**	**2281**			**2279 + 166**
	No bikes	**323**	**144**	**133**			**67**
	No parking	**174**	**106**	**97**			**6**
100	Satisfied	**3213**	**3401**	**3437**			**4165 + 424**
	No bikes	**1385**	**1230**	**1210**			**554**
	No parking	**618**	**570**	**556**			**32**
250	Satisfied	**8720**	**8870**	**8880**			—
	No bikes	**3615**	**3495**	**3481**			—
	No parking	**1464**	**1423**	**1426**			—

Evaluation of the system

Result: Medium case demand - edoras

Size	Indicator	Empty	OB	OB-corr	DP	CC	OT
20	Satisfied	**1130**	**1649**	**1666**			**1556 + 288**
	No bikes	**660**	**210**	**191**			**97**
	No parking	**188**	**104**	**104**			**23**
50	Satisfied	**3849**	**4173**	**4234**			**4330 + 311**
	No bikes	**973**	**714**	**661**			**397**
	No parking	**269**	**198**	**185**			**44**
100	Satisfied	**5290**	**5589**	**5641**			**7096 + 790**
	No bikes	**4283**	**4021**	**3970**			**2385**
	No parking	**918**	**863**	**862**			**128**
250	Satisfied	**12817**	**13024**	**13051**			–
	No bikes	**13058**	**12863**	**12830**			–
	No parking	**2161**	**2132**	**2137**			–

Evaluation of the system

Result: High case demand - edoras

Size	Indicator	Empty	OB	OB-corr	DP	CC	OT
20	Satisfied	**1535**	**2333**	**2393**			**2364 + 457**
	No bikes	**1579**	**825**	**762**			**430**
	No parking	**219**	**144**	**145**			**45**
50	Satisfied	**5845**	**6364**	**6436**			**6533 + 468**
	No bikes	**2309**	**1850**	**1782**			**1391**
	No parking	**353**	**260**	**250**			**74**
100	Satisfied	**6977**	**7392**	**7482**			**9858 + 1193**
	No bikes	**9588**	**9188**	**9096**			**6176**
	No parking	**1113**	**1073**	**1072**			**242**
250	Satisfied	**15285**	**15560**	**15537**			—
	No bikes	**29447**	**29173**	**29193**			—
	No parking	**2364**	**2343**	**2348**			—

Some remarks for the pricing experiments

Update of the prices: all **15** minutes

Walk/Bike: a travel of **x** seconds by bike 'costs' **5x** seconds when done on foot

The prices in simulator: in seconds. For instances with size

- **20**: maximal price: **3000**
- **50**: maximal price: **4200**
- **100**: maximal price: **6000**

To make the conversion, take (value of travel time in a Western city)

$$\mathbf{8} \text{ euros} = \mathbf{1} \text{ hours}$$

A web-site where the simulator can be downloaded:

http://cermics.enpc.fr/∼meuniefr/OADLIBSim_Site/index.html