Bike sharing and operations research

Frédéric Meunier
Cermics, Ecole des Ponts

19 février 2019

History - first generation

No docks.

- 1965: White Bicycle Plan, in Holland. No stations, no fees. Complete disaster.
- 1974: Vélos Jaunes, in La Rochelle. Relative success. Still working (after adaptations).
- 1993: Green Bicycle Scheme, in Cambridge. Stations but no fees. Complete disaster.
- ...

History - second generation

Docks and information technology.

- 1995: Grippa, Portsmouth, for students. Abandoned in 1998 (too expensive and not efficient).
- 1995: ByCykler, Copenhague, abandonded in 2012 (for financial reasons).
- 1997: Grippa, Rotterdam. Abandoned in 1998 ; poorly functioning electronic bike racks.
- 1997: CityBikes, Helsinki. Abandoned in 2010, too high vandalism.
- 2001: BiCyBa, Bratislava. After three months, all bikes had been stolen.

History - Successes

- 2005: Vélov', Lyon. Now: 350 stations and 3'000 bikes.
- 2007: Bicing, Barcelona. Now: 400 stations and 6'000 bikes.
- 2007: Vélib, Paris. Now: 1'200 stations and 16'000 bikes.
- 2008: Hangzhou Public Bicycle, Hangzhou. Now: 2'700 stations and 84'000 bikes.
- 2009: Bixi, Montreal. Now: 300 stations and 3'000 bikes.
- 2010: Capital Bikeshare, Washington D.C. Now: 288 stations and 2'800 bikes.

Examples

Examples

Limitations

⋆ No impact on urban congestion and $CO2$ emissions.

In Paris every day:

- 80’000 trips by Vélib’
- 4’049’000 trips by metro
- 2’700’000 trips by RER
- 2’000’000 by car.

⋆ Vandalism.

⋆ Never profitable.

⋆ Quality of service: the full/empty station problem.

Limitations

⋆ No impact on urban congestion and CO2 emissions.

In Paris every day:

- 80’000 trips by Vélib’
- 4’049’000 trips by metro
- 2’700’000 trips by RER
- 2’000’000 by car.

⋆ Vandalism.

⋆ **Never profitable.**

⋆ **Quality of service: the full/empty station problem.**

Typical OR problems in bike sharing systems

- Station location
- Fleet dimensioning
- Inventory setting
- Rebalancing incentives
- Bike repositioning

Except the first problem, they all have static/dynamic versions.

Station location

Nair-Miller-Hooks model (2014)

Possible stations: P; OD pairs in the city: K

Objective. Maximize the revenue

Variables. $\begin{cases} x_i : & \text{binary variable for opening station } i \in P \\ y_i : & \text{number of parking docks in station } i \in P \\ z_i : & \text{initial number of bikes in station } i \in P \\ v_{ijk} : & \text{number of users connecting } k \in K \text{ via } i \to j. \end{cases}$

Constraints. Initial budget, logical constraints between $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$, reaction of the users.

$$\text{Model is} \begin{cases} \text{Min} & F(\boldsymbol{v}) \\ \text{s.t.} & G(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \leqslant \boldsymbol{0} \\ & \boldsymbol{v} = \text{UserReaction}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \\ & \boldsymbol{x} \in \{0,1\}^P, \boldsymbol{y}, \boldsymbol{z} \in \mathbb{Z}_+^P, \boldsymbol{v} \in \mathbb{R}_+^{P \times P \times K} \end{cases}$$

with F and G linear.

Nair-Miller-Hooks model, cont'd

$$\left\{ \begin{array}{ll} \text{Min} & F(\boldsymbol{v}) \\ \text{s.t.} & G(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \leqslant \boldsymbol{0} \\ & \boldsymbol{v} = \text{UserReaction}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \\ & \boldsymbol{x} \in \{0,1\}^P, \boldsymbol{y}, \boldsymbol{z} \in \mathbb{Z}_+^P, \boldsymbol{v} \in \mathbb{R}_+^{P \times P \times K} \end{array} \right.$$

Spiess and Florian (1989):

$$\boldsymbol{v} = \text{UserReaction}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$$

$$\Longleftrightarrow$$

$\boldsymbol{v}$ is an optimal solution of a linear program

$$\left\{ \begin{array}{ll} \text{Min} & f(\boldsymbol{v}, \boldsymbol{w}) \\ \text{s.t.} & g_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}}(\boldsymbol{v}, \boldsymbol{w}) \leqslant \boldsymbol{0} \\ & \boldsymbol{v} \in \mathbb{R}_+^{P \times P \times K}, \boldsymbol{w} \in \mathbb{R}_+^{P \times K} \end{array} \right.$$

Thus, the Nair-Miller-Hooks model relies on bilevel programming.

Solving the Nair-Miller-Hook model

Optimality conditions of a convex program (KKT conditions) can be linearized with binary variables.

Nair-Miller-Hooks model can thus be rewritten as a MIP.

Tractable only for $|P| \leqslant 40$.

Optimal design often inefficient for users!
Subsidies for operators?

Fleet dimensioning

George-Xia model (2011)

- Users arrive at station i according to a Poisson process of rate λ_i and choose a destination station with probability p_{ij}.
- Duration of a trip ij follows a general distribution of finite mean, depending only on the pair i, j.
- No repositioning performed by the operator.
- Stations have infinite capacity.

Availability of station i: probability of finding a bike for a user arriving at station i

Objective.

⋆ Determine availability at steady state for each station as a function of the total number n of bikes.

⋆ Find the number of bikes maximizing operator's profit (rents, penalties for users not finding bikes, maintenance costs).

Availability

Modeling via closed queueing networks:

- Customers are bikes
- Station i is a server with λ_i as service rate
- Trip ij is a server with service time distributed according to trip duration distribution.

General theory of closed queueing networks $\rightarrow$ closed-form formula for steady-state availability $A_i(n)$ (in function of the total number n of bikes).

Computation of the formula very expensive, but general principles can be derived.

Maximizing profit

$\text{Max}_{n \in \mathbb{Z}_+} \sum_{ij} r_{ij} L_{ij}(n) - \sum_i p_i \lambda_i (1 - A_i(n)) - cn$

- r_{ij} per-unit time income for a trip ij
- p_i penalty incurred when a user does not find a bike at station i
- c per-unit time maintenance cost
- $L_{ij}(n)$ number of bikes doing trip ij at steady state (hard to compute)

They proved:
The objective function is concave.

They proposed heuristics and evaluated them experimentally.

Inventory setting

Nair-Miller-Hooks model (2011)

- Each station i has a capacity C_i and an initial number V_i of bikes
- Moving bikes from station i to station j costs $a_{ij} + \delta \times$ number of bikes
- Random demand ζ_i^b in bikes at station i over the whole period (with known distribution)
- Random demand ζ_i^d in free parking docks at station i over the whole period (with known distribution)
- Demand must be satisfied with probability p

Objective. Minimize the total cost.

Nair-Miller-Hooks model, cont'd

$$
\begin{aligned}
\text{Min} \quad & \sum_{i,j \in S} (a_{ij} x_{ij} + \delta y_{ij}) \\
\text{s.t.} \quad & \\
& \mathbb{P} \begin{pmatrix} V_i + \sum_{j \in S} (y_{ji} - y_{ij}) + \zeta_i^d \geqslant \zeta_i^b, & i \in S \\ C_i - V_i + \sum_{j \in S} (y_{ij} - y_{ji}) + \zeta_i^b \geqslant \zeta_i^d, & i \in S \end{pmatrix} \geqslant p \\
& \sum_{j \in S} y_{ij} \leqslant V_i & i \in S \\
& \sum_{j \in S} y_{ji} \leqslant C_i - V_i & i \in S \\
& y_{ij} \leqslant M x_{ij} & i, j \in S \\
& y_{ij} \geqslant 0 \text{ and integer}, \; x_{ij} \in \{0, 1\} & i, j \in S,
\end{aligned}
$$

where S is the set of stations.

Methods and results

p-efficient points of Prékopa (1990).

$\boldsymbol{b}$ is a p-efficient point if

$$A\boldsymbol{x} \geqslant \boldsymbol{b} \quad \Longrightarrow \quad \mathbb{P}(A\boldsymbol{x} \geqslant \zeta) \geqslant p$$

and is nondominated for this property.

For integer variables, the number of p-efficient points is finite $\longrightarrow$ enumeration algorithm

- p-efficient points can be precomputed.
- Methods validated through simulations on a car sharing system with 14 stations, 202 parking docks in total, and 94 cars. (Real-data from Singapore.)
- Able to ensure $p \geqslant 0.9$ for finding bikes, and $\simeq 0.6$ for finding docks, by computing regularly repositioning (1 hour computation)

Raviv-Kolka model (2013)

- A single station with a capacity C.
- Non-homogeneous Poisson process for the demands in bikes and parking docks.

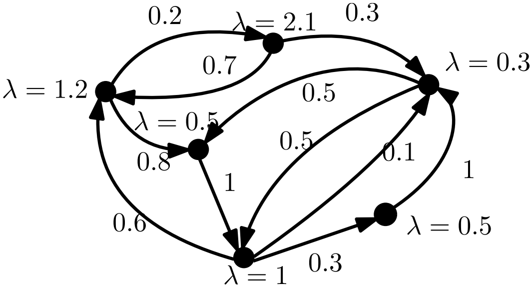
Objective. Find the initial inventory I_0 such that the total number of unsatisfied users is minimized over a given period.

They proved:
This number is a convex function in I_0.

Easy to solve for a real process, and used in practice in Tel-Aviv for a while.

Interaction between the stations

Limitation of the previous two methods: interaction between stations is neglected.



Infinite capacity for the stations, n bikes, T time steps

Location of the n bikes maximizing expected number of users finding a bike?

To author's knowledge, this question has not been investigated yet.

The real problem

- Bike-sharing system with Poisson processes
- Stations with finite capacity
- No repositioning performed by the operator

Objective. Find the initial inventories of the stations so that the expected number of unsatisfied users is minimized over a given horizon.

Dasner, Raviv, Tzur, and Chemla (2015)

Method. Local search with simulations to estimate the quality of the solution; surrogate functions to reduce the size of the neighborhoods.

Able to provide provably good solutions (quality proved by simulation) for real networks in about 1 hour.

On real-data, decrease excess time

- by more than 60% with respect to loading each station to half its capacity.
- by 0.9% to 9% with respect to the Raviv-Kolka model.

Rebalancing incentives

Static and dynamic incentives

In Paris or Lyon, elevated stations are “bonus”: leaving a bike at such a station leads to a gain for the user.

This is static incentive, implemented in various places.

Ubiquitous idea, but never implemented: incentives depending on the current state of the system.
$\rightarrow$ Dynamic incentives

Chemla et al. (2013)

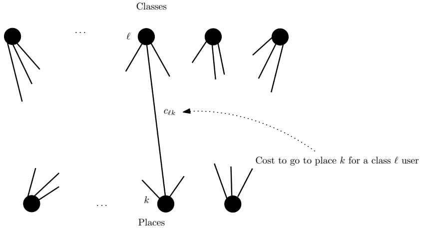
Model.

- There is a price p_k for leaving a bike at station k.
- Every 15 minutes, prices are updated.

Objective. Maintain each station close to a target level.

Chemla et al. (2013, cont'd)

Dynamic incentive based on Monge market problem.



Class ℓ has a_ℓ customers, each customer wants to go to some place, each place k has a capacity b_k.

Objective. Fix prices p_k for access to place k so that no more than b_k customers go to place k.

Solution. $p_k =$ dual variable of place k (in Monge transportation problem with capacities a_ℓ and b_k)

Chemla et al. (2013), cont'd

- Classes are pairs (i, j) (users wanting to do $i \to j$).
- Places are stations
- Capacity b_k is the number of missing bikes.
- Number of users in class (i, j): expected number of trips $i \to j$ per unit time.

Since it is a Monge transportation problem, prices p_k are computed via a simple linear program (without integer variables).

Chemla et al. (2013), cont'd

- Capacity of a station: 20
- Target level: 14
- The mean average rate is 5 users / minute.

For 100 users:

Size	Status	Empty	1SH	1SHF	2SH	2S1SHF	Pricing
	Satisfied	70	99	99	98	98	84
20	No bike	12	1	1	0	1	0
	No parking dock	18	0	0	2	1	0
	Rejection						16
	Satisfied	62	67	67	66	67	81
100	No bike	26	23	23	23	22	10
	No parking dock	12	10	10	11	11	0
	Rejection						9

Maximum prices met by the system: 7€ for 20 stations and 16€ for 100 stations.

Two other dynamic pricing approaches

Pfrommer, Warrington, Schildbach, Morari (2014).

- Pricing mechanism combined with truck operating.
- Prices are of the form $p_{k,k'}$: target station k, alternative stations k'.
- Update via a quadratic program, minimizing dissatisfaction and operating costs.
- Suppose to have a good predictive model running in real-time.
- Claim 87% service level on weekends for London system (via simulation) without trucks operating.

Singla, Santoni, Bartók, Mukerji, Meenen, Krause (2015).

- Prices are independent of the state.
- Assume that there is a inherent price for accepting an alternative station no too far from the target station.
- Price updates are due to a simple learning mechanism.
- No target level for the stations.

Bike repositioning

Static and dynamic repositionings

Two types of repositioning are usually considered.

Static.

- No users in the system.
- Bikes are moved between stations so that the repartition of the bikes is close to a predetermined one.
- Models system during night: idle, preparation of morning rush.

Dynamic.

- Users in the system.
- Bikes are moved to avoid problematic situations.
- Models system during the day.

Static repositioning

Raviv, Tzur, Forma (2013)

- Fleet of trucks moving bikes between stations.
- User dissatisfaction in a station: piecewise convex function depending on the inventory.
- Objective function: tradeoff between total user dissatisfaction and routing costs.
- Various direct integer programs.

Schuijbroek, Hampshire, van Hoeve (2017)

- Each station has an target interval, in which the inventory must lie.
- Objective function: minimize makespan.
- Cluster-first route second heuristic.
- Clustering problem simultaneously considers inventory level feasibility and approximate routing costs.
- Experiments show good behaviors (find solutions at $10\% - 20\%$ in 5 min, for $2 - 5$ trucks and about 25 stations).
- Used in practice?

An academic problem

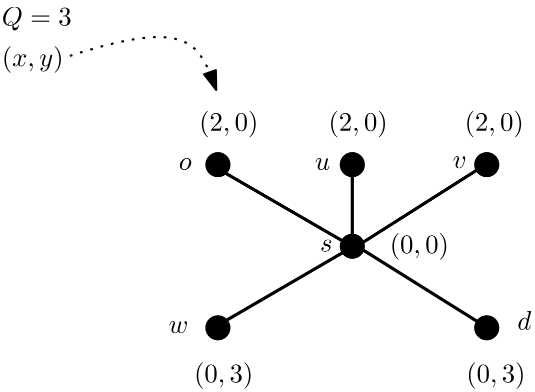
Extension of 1-PDTSP by Hernández-Pérez and Salazar-González (2004)

Input. Directed graph $G = (V, A)$, origin and destination $o, d \in V$, initial and target bike repartitions $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{Z}_+^V$, arc cost $\boldsymbol{c} \in \mathbb{R}_+^A$, truck capacity Q, station capacity C

Output. A route with pickup and delivery operations, of minimal cost, making the system go from $\boldsymbol{x}$ to $\boldsymbol{y}$.

Preemption: Temporary storage at intermediate stations is allowed.

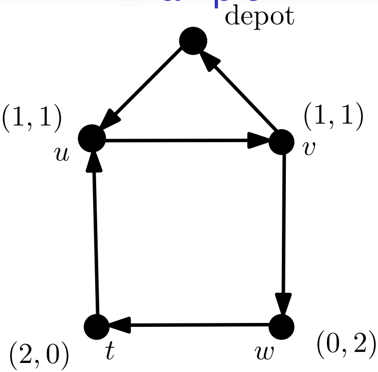
Example



With no preemption: best solution in 10 moves.

With preemption: best solution in 8 moves.

Example



Because of preemption, optimal solutions may visit already balanced stations and also visit arcs between stations with nonegative imbalance several times.

(This holds also for the complete graph obtained by metric completion.)

Theoretical features

- ⋆ NP-hard
- ⋆ Polynomial $31/3$-approximation
- ⋆ Polynomial algorithm for trees when station capacity $C = +\infty$
- ⋆ Open complexity for cycles.

Results obtained with undergraduate students in 2011.

Trees

For any edge e in a tree T, the set U_e is an arbitrarily chosen connected component of $T \setminus e$.

Theorem

When G is a tree and $C = +\infty$, in any optimal solution, every edge e is traversed

$$\left\lceil \frac{\sum_{v \in U_e} (x_v - y_v)}{Q} \right\rceil$$

times, plus or minus one move, depending on the respective positions of o, d, and e.
Moreover, there is a simple greedy procedure building an optimal solution.

Trees, cont'd

More precisely:

Theorem

When G is a tree and $C = +\infty$, in any optimal solution, every edge e is traversed

$$\max\left(\left\lceil\frac{\sum_{v\in U_e}(x_v - y_v)}{Q}\right\rceil + \eta_e(o,d), \mu_e(o,d,\boldsymbol{x},\boldsymbol{y})\right)$$

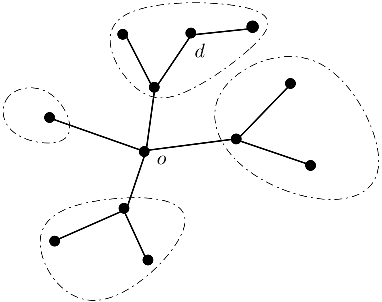
times.
Moreover, there is a simple greedy procedure building an optimal solution.

$\eta_e(o,d) \in \{-1, 0, 1\}$ and $\mu_e(o,d,\boldsymbol{x},\boldsymbol{y}) \in \{0, 1, 2\}$.

Greedy algorithm for trees

Input. Tree T, repartitions $\boldsymbol{x}$, $\boldsymbol{y}$, truck capacity Q, initial and target positions o, d
Output. First action of the truck

if *There is a connected component U with bike excess* **then**
  – Upload all bikes on o;
else
  – Choose a connected component U not in target state (and, if possible, not containing d);
  – Load bikes from o until truck load reaches $\min\left(Q, \sum_{v \in U}(y_v - x_v)\right)$;
end
– Enter U;



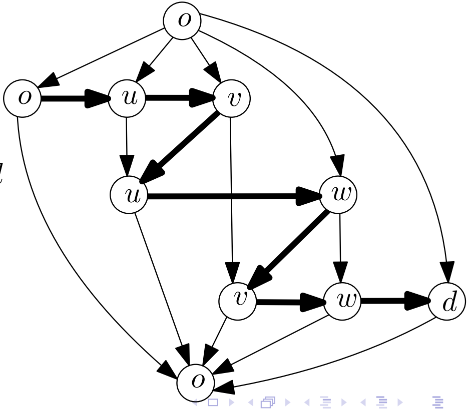
Combinatorial encoding

For a general graph:

Theorem (Chemla, M., Wolfler Calvo)

Computing the best loading and unloading operations for a given sequence of visited stations reduces to a maximum flow problem (strongly polynomial).

Sequence: o, u, v, u, w, v, w, d



Combinatorial encoding, cont'd

Best: If the station capacities are $+\infty$, then it means that we get a final repartition $\boldsymbol{y}'$ minimizing $\|\boldsymbol{y} - \boldsymbol{y}'\|_1$.

Otherwise, we get $\boldsymbol{x}', \boldsymbol{y}'$, maximizing $\sum x'_v$, such that $\boldsymbol{x}' \leqslant \boldsymbol{x}$, $\boldsymbol{y}' \leqslant \boldsymbol{y}$, and the sequence is a feasible solution for the repartition $\boldsymbol{x}', \boldsymbol{y}'$.

In particular, if there are feasible loading/unloading operations compatible with the sequence, the algorithm finds them.

Local search

Cruz, Subramanian, Bruck, Iori (2017) designed an efficient iterated local search based on this combinatorial encoding.

Max flow algorithm used to evaluate quality of current solution

Able to deal with unfeasible route.

Experimental results

Instances adapted from 1-PDTSP by Hernández-Pérez and Salazar-González

Typical performance: less than 5 seconds to find an optimal solution for :

- Station capacity: 20
- Number of stations: 40
- Total number of bikes: 400
- Truck capacity: 30

Lower bound

$$
\begin{aligned}
\text{Min} \quad & \sum_{a \in A} c_a z_a \\
\text{s.t.} \quad & \sum_{a \in \delta^+(v)} z_a = \sum_{a \in \delta^-(v)} z_a & \forall v \in V \\
& \sum_{a \in \delta^+(X)} z_a \geqslant \max\left(\left\lceil \frac{|\sum_{v \in X}(x_v - y_v)|}{Q} \right\rceil, \mu(\overline{X}) \right) & \forall X \subseteq V \\
& z_a \in \mathbb{Z}_+ & \forall a \in A
\end{aligned}
$$

$$
\mu(\overline{X}) = \begin{cases} 1 & \text{if there is an unbalanced vertex in } \overline{X} \\ 0 & \text{otherwise.} \end{cases}
$$

Erdoğan, Battara, Wolfler Calvo (2015) solve by branch-and-cut up to 60 vertices

Other versions

Nonpreemptive version. Seems to be harder

With many trucks. Even solving instances with 20 stations to optimality is challenging

Dynamic repositioning

Pfrommer, Warrington, Schildbach, Morari (2014)

- Planning horizon 40 minutes
- Each truck visits at most 4 stations over the horizon
- Simultaneously, heuristic builds promising routes and quadratic program computes the actions (for next planning horizon).
- Once routes are computed, they are fixed.
- Assume good prediction on the demand.
- Evaluated through simulation.

Future work or “Open questions”

Initial inventory

- Arrival rates: λ_i
- "Probabilities of destination": p_{ij}
- Travel times: t_{ij}
- No truck operating.

Objective. Find the initital inventories maximizing expected number of satisfied users over a planning horizon.

Already with station capacity $C = +\infty$: almost nothing is known.

Dynamic routing

- Arrival rates: λ_i
- "Probabilities of destination": p_{ij}
- Bike travel times: t_{ij}
- Truck travel times: τ_{ij}
- One truck operating of capacity Q

Objective. Maximize the average number of satisified users per unit time.

Already with station capacity $C = +\infty$: almost nothing is known.

Circle

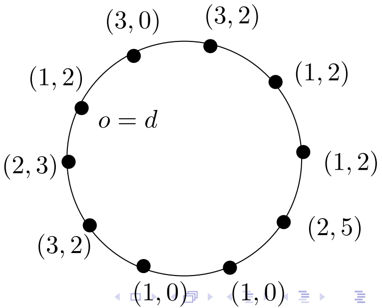
Input. Bidirected cycle $G = (V, A)$, origin and destination $o, d \in V$, initial and target bike repartitions $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{Z}_+^V$, truck capacity Q, station capacity C

Output. A route with pickup and delivery operations, of minimal length, making the system go from $\boldsymbol{x}$ to $\boldsymbol{y}$.

Preemption: Temporary storage at intermediate stations is allowed.

Is this problem polynomial?

Open even for $Q = C = +\infty$.



References I

- M. Benchimol, P. Benchimol, B. Chappert, A. de la Taille, F. Laroche, F. Meunier, and L. Robinet, *Balancing the stations of a self service "bike hire" system*, RAIRO - Operations Research **45** (2011), 37–61.
- D. Chemla, F. Meunier, T. Pradeau, R. Wolfler Calvo, and H. Yahiaoui, *Self-service bike sharing systems: simulation, repositioning, pricing*, Tech. report, 2013.
- D. Chemla, F. Meunier, and R. Wolfler Calvo, *Bike-sharing systems: solving the static rebalancing problem*, Discrete Optimization **10** (2013), 120–146.
- F Cruz, A. Subramanian, B.P. Bruck, and M. Iori, *A heuristic algorithm for a single vehicle static bike sharing rebalancing problem*, Computers & Operations Research **79** (2017), 19–33.
- G. Erdoğan, M. Battarra, and R. Wolfler Calvo, *An exact algorithm for the static rebalancing problem arising in bicycle sharing systems*, European Journal of Operational Research **245** (2015), 667–679.
- D.K. George and C.H. Xia, *Fleet-sizing and service availability for a vehicle rental system via closed queueing networks*, European Journal of Operational Research **211** (2011), 198–207.
- H. Hernandez Pérez and J.J. Salazar González, *A branch-and-cut algorithm for a traveling salesman problem with pickup and delivery*, Discrete Applied Mathematics **145** (2004), 126–139.

References II

- Gilbert Laporte, Frédéric Meunier, and Roberto Wolfler Calvo, *Shared mobility systems*, 4OR **13** (2015), 341–360.
- R. Nair and E. Miller-Hooks, *Fleet management for vehicle sharing operations*, Transportation Science **45** (2011), 524–540.
- ______, *Equilibrium network design of shared-vehicle systems*, European Journal of Operational Research **235** (2014), 47–61.
- J. Pfrommer, J. Warrington, G. Schildbach, and M. Morari, *Dynamic vehicle redistribution and online price incentives in shared mobility systems*, IEEE Transactions on Intelligent Transportation Systems **99** (2014), 1–12.
- T. Raviv and O. Kolka, *Optimal inventory management of a bike-sharing station*, IEEE Transactions **45** (2013), 1077–1093.
- T. Raviv, M. Tzur, and I.A. Forma, *Static repositioning in a bike-sharing system: Models and solution approaches*, EURO Journal on Transportation and Logistics **2** (2013), 187–229.
- H. Spiess and M. Florian, *Optimal strategies: A new assignment model for transit networks*, Transportation Research Part B **23** (1989), 83–102.
- J. Schuijbroek, R. Hampshire, and W.-J. van Hoeve, *Inventory rebalancing and vehicle routing in bike sharing systems*, European Journal of Operational Research (2013), 992–1004.

References III

A. Singla, M. Santoni, G. Bartók, P. Mukerji, M. Meenen, and A. Krause, *Incentivizing users for balancing bike sharing systems*, Proceedings of Conference on Artificial Intelligence (AAAI), 2015, pp. 723–729.

Thank you.