



Vehicle Sharing System Optimization: Scenario-based approach

Ariel Waserhole, Vincent Jost, Nadia Brauner

► To cite this version:

Ariel Waserhole, Vincent Jost, Nadia Brauner. Vehicle Sharing System Optimization: Scenario-based approach. 2012. hal-00727040v2

HAL Id: hal-00727040

<https://hal.science/hal-00727040v2>

Preprint submitted on 14 Nov 2012 (v2), last revised 31 May 2013 (v4)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Vehicle Sharing System Pricing Regulation: Deterministic Approach, Complexity results

Ariel Waserhole^{1,2}

Vincent Jost²

Nadia Brauner¹

¹ G-SCOP, UJF Grenoble

² LIX CNRS, École Polytechnique Palaiseau

November 14, 2012

Abstract

This paper gives complexity results on the deterministic versions of the Vehicle Sharing System Pricing problem. Solving these deterministic problems could give some Upper Bound on the general problem or help designing robust optimization algorithm. However we show that most of them are NP-hard.

1. We define the First Come First Serve (FCFS) flow, and the priced FCFS flow based on a scenario approach.
2. We define two deterministic problems based on the priced FCFS flow: MAX FCFS FLOW TRIP PRICING AND MAX FCFS FLOW STATION PRICING. WE SHOW THAT THEY ARE BOTH NP-HARD.
3. WE DEFINE THE MAX FCFS FLOW STATION CAPACITIES PROBLEM BASED ON THE FCFS FLOW AND SHOW ITS NP-HARDNESS.
4. WE TALK ABOUT THE MAX FLOW UPPER BOUND ON FCFS FLOW PROBLEMS.

1 Introduction

1.1 Context

Shoup (2005) reports that, based on a sample of 22 US studies, cars looking for a parking spot contribute to 30% of the city traffic. Moreover cars are used less than 2 hours per day on average but still occupy a parking spot the rest of the time! Could we have less vehicles and satisfy the same demand level?

Recently, the interest in Vehicle Sharing Systems (VSS) in cities has increased significantly. Indeed, urban policies intend to discourage citizens to use their personal car downtown by reducing the number of parking spots, street width, etc. VSS seem to be a promising solution to

reduce jointly traffic and parking congestion, noise, and air pollution (proposing bikes or electric cars). They offer personal mobility allowing users to pay only for the usage.

We are interested in short-term one-way VSS where vehicles can be taken and returned at different places (paying by the minute). Associated with classical public transportation systems, short-term one-way VSS help to solve one of the most difficult public transportation problem: the last kilometer issue (DeMaio, 2009). This is not the case for round-trip VSS where vehicles have to be returned at the station where they were taken.

The first large-scale short-term one-way VSS was the bicycle VSS Vélib'. It was implemented in Paris in 2007 and now has more than 1200 stations and 20 000 bikes selling around 110 000 trips per day. It has inspired several other cities all around the world; Now more than 300 cities have such a system, including Montréal, Beijing, Barcelona, Mexico City, Tel Aviv (DeMaio, 2009).

1.2 One-way Vehicle Sharing Systems: a management issue

However if freedom increases for the user in the one way model, it implies a higher complexity in its management. In round trip type rental systems, the only stock that is relevant when managing yield and reservations is the number of available vehicles. In one-way systems, a new problem occurs since vehicles aren't the only key resource any more. In practice, parking stations have a maximum number of spots, and when the total number of vehicle is comparable to the total number of parking spots, available parking spots become a new key resource.

Since first bicycle VSS, problems of bikes and parking spots availability have appeared very often. Reasons are various but we can highlight two important phenomenon: the gravitational effect which indicates that a station is constantly unbalanced (as Montmartre hill in Vélib'), and the tide phenomenon representing the oscillation of demand intensity along the day (as morning and evening flows between working and residential areas).

To improve the efficiency of the system, in the literature, different perspectives are studied. At a strategic level, some authors consider the optimal capacity and locations of bike rental stations. Shu et al. (2010) propose a stochastic network flow model to support these decisions. They use their model to design a bicycle VSS in Singapore based on demand forecast derived from current usage of the mass transit system. Lin and Ta-Hui (2011) consider a similar problem but formulate it as a deterministic mathematical model. Their model is aware of the bike path network and mode sharing with other means of public transportation.

At a tactical level, other authors investigate the optimal number of vehicles given a set of stations. George and Xia (2011) study the fleet sizing problem with constant demand and no parking capacity. Fricker and Gast (2012); Fricker et al. (2012) look into the optimal sizing of a fleet in "toy" cities, where demand is constant over time and identical for every possible trip, and all stations have the same capacity $\mathcal{K}$. They show that even with an optimal fleet sizing in the most "perfect" city, if there is no operational system management, there is at least a probability of $\frac{2}{\mathcal{K}+1}$ that any given station is empty or full.

At an operational level, in order to be able to meet the demand with a reasonable standard of quality, in most bicycle VSS trucks are used to balance the bikes among the stations. The problem is to schedule vehicle routes to visit some of the stations to perform pickup and delivery so as to minimize the number of users who cannot be served, *i.e.*, the number of users who try to take a bike from an empty station or to return it to a full station. In the literature many papers deal already with this problem. A static version of the bicycle VSS balancing problem is treated in Chemla et al. (2011) and a dynamic one in Contardo et al. (2012).

A new type of VSS has appeared lately: one-way Car VSS with Autolib' in Paris and Car2go in more than 10 cities (Vancouver, San Diego, Lyon, Ulm...). With cars, operational balancing optimization through relocation seems inappropriate due to their size. We have to find another way to optimize the system.

1.3 Regulation through pricing

The origin of Revenue Management (RM) lies in airline industry. It started in the 1970s and 1980s with the deregulation of the market in the United States. In the early 1990s RM techniques were then applied to improve the efficiency of round trip Vehicle Rental Systems (VRS), see Carroll and Grimes (1995) and Geraghty and Johnson (1997). One way rental is now offered in many VRS, however as one can see in practice for car VRS that it is always much more expensive than round trip rental. We haven't found in the literature authors tackling the one way VRS RM problem. We can only cite Haensela et al. (2011) that model a network of only round trip car VRS but with the possibility of transferring cars between rental sites for a fixed cost. For trucks rental on the contrary, companies such as Rentn'Drop in France or Budget Truck Rental in the United States are specialized in the one way rental offering dynamic pricing. This problem is tackled by Guerriero et al. (2012) that consider the optimal managing of a fleet of trucks rented by a logistic operator, to serve customers. The logistic operator has to decide whether to accept or reject a booking request and which type of truck should be used to address it.

Anyway results for one way VRS are not directly applicable to VSS, because they differ on several points: 1) Renting are by the day in VRS and by the minute in VSS with a possible high intensity; 2) One way rental is the core in VSS, for instance only 5% of round trip rental in Bixi (Morency et al., 2011), and it is classically the opposite in car VRS. 3) There is usually no booking in advance in VSS, it is a first come first serve rule, whereas usually trips are planned several days in advance in VRS.

In this paper we are looking at VSS and optimization through pricing. Assuming that demand is elastic, we want to use prices to influence user choices in order to drive the system towards its most efficient dynamic.

This work is part of a preliminary study using operation research to 1) Establish the interest of VSS pricing regulation system 2) Give good and possibly simple pricing policies for the operational management.

A model for the Stochastic Vehicle Sharing System Pricing problem based on Markov Decision

Processes (MDP) is proposed in Waserhole et al. (2012). The number of states of this MDP formulation is exponential and exact solution techniques are therefore not applicable. Hence they give a fluid approximation to compute static policies.

1.4 Deterministic approach

In this paper we develop a deterministic model for the Vehicle Sharing System Pricing problem. It is a classic and natural way to study deterministic version of stochastic problems. We consider that all trip requests are available at the beginning of the time horizon. In practice it amounts to optimize a posteriori the system, using exploitation data (scenario of the day).

On the one hand, such optimization gives a bound for an on-line stochastic optimization problem on a given instance. On the other hand, solving efficiently the deterministic problem on a scenario is the first step toward robust optimization methods, at least for the models describing uncertainty by sets of scenarii. Moreover Morency et al. (2011) showed that in Montreal BSS Bixi 68% of the trips were made by “members”, and that their frequencies of use are quite stable along the week. We can hence consider that this demand is quasi deterministic and considering deterministic requests might be a good approximation.

In the remaining of the paper we derive complexity results for the deterministic Vehicle Sharing System Pricing problem. We show that the combinatorial problems involved are NP-hard.

2 First Come First Serve constrained flows

We define a new type of constrained flow, the “First Come First Serve flow” (FCFS flow) over a time and space network. Even if it is not explicitly specified nor named, this constraint is already known in continuous time models. For instance it arises naturally in many applications such as in the fluid approximation of a Markov Decision Process (Waserhole et al., 2012). However to the best of our knowledge it is not studied, nor mentioned, in a discrete time model.

In order to remain in the lexical field of VSS we speak about a flow of vehicles transiting (thanks to users) along stations. In a more general context it can be called a flow of resources along locations.

2.1 First Come First Serve flows in Time and Space Network

Consider a automatic flow, that is without decision variables, of N vehicles transiting among a set S of stations S with capacities $\mathcal{K}_s$, $s \in S$ on a time horizon $H = [0, T]$. At time 0 we have an initial repartition of the vehicles among the stations. The vehicles then transit between stations by accepting spatio-temporal user trip requests. The set of trip requests R is known at

the beginning of the horizon. A trip request $r \in R$ asks for a vehicle between an original station s_o^r at time t_o^r and a destination station s_d^r at time t_d^r for a price p^r .

Given such data we build a time and space network and follow the evolution of the process. From the beginning of the horizon, we increase the time until an event (trip request or vehicle arrival) occurs. At time t the trip request $r = (s_o^r, t_o^r = t, s_d^r, t_d^r, p^r) \in R$ is accepted if and only if there is a vehicle available at station s_o^r at time t and a parking spot available at station s_d^r also at time t . If the trip request r is accepted: It generates a gain p^r , removes a vehicle from station s_o^r and reserves a parking spot at station s_d^r until time $t_d^r \geq t$; At time t_d^r the vehicle is available again at station s_f^r . If the trip is rejected, nothing happens.

We call this deterministic process “First Come First Serve Flow”. The gain generated by this process can be evaluated in linear time. Figure 1 gives an example of a First Come First Serve flow with three stations, 12 requests and 1 vehicle available at station a at the beginning of the horizon. In this “scenario”, only 3 trips among 12 are served giving a gain of 24.

Remark 1 *FCFS flows can be evaluated in linear time. Hence the decision versions of the optimization problems considered in this paper are in NP.*

2.2 Priced First Come First Serve flows

Consider now that the system has established a fixed price $p_{a,b}$ for each trip $(a,b) \in S^2$. The price (denoted $p_{\max}^r$) of each request $r \in R$ is now to be interpreted as the maximum amount the user is willing to pay for taking the trip. Let R_p be the requests which accept to pay the asked prices: $R_p = \{r \in R : p_{\max}^r \geq p_{s_o^r, s_d^r}\}$. A request $r \in R$ to take a trip from station s_o^r to station s_d^r is accepted if and only if it can afford the asked price, *i.e.* $p_{\max}^r \geq p_{s_o^r, s_d^r}$ and there is a vehicle and a parking spot available at time t_o^r . When request r is accepted, it generates a gain $p_{s_o^r, s_d^r}^r$ (not $p_{\max}^r$).

We call this automatic process “Priced First Come First Serve Flow”. It amounts in evaluating the gain generated by a First Come First serve flow with requests R_p induced by a set of prices $p : S^2 \rightarrow \mathbb{R}$ and can also be evaluated in linear time.

Figure 2 gives an example of an evaluation of such a process with 3 stations and 1 vehicle. Here, out of 12 requests, 10 can afford the asked price and finally 6 are served for a gain of 49.

3 Pricing problems

We now define optimization problems based on the priced First Come First flow. We want to optimize the system through price leverage: We are looking for pricing policies that maximize the gain of the induced priced FCFS flow.

We state the complexity of two optimization problems both shown NP-Hard: The FCFS flow trip pricing problem which sets a price for every trips independently and the FCFS flow station

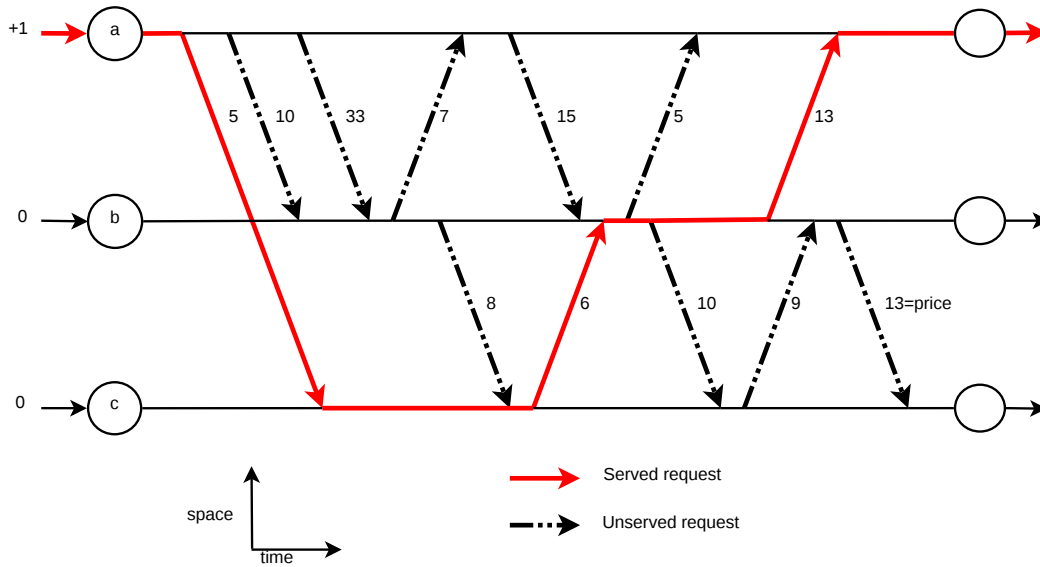


Figure 1: First Come First Serve Flow with value 24. Trip prices are on the arrows.

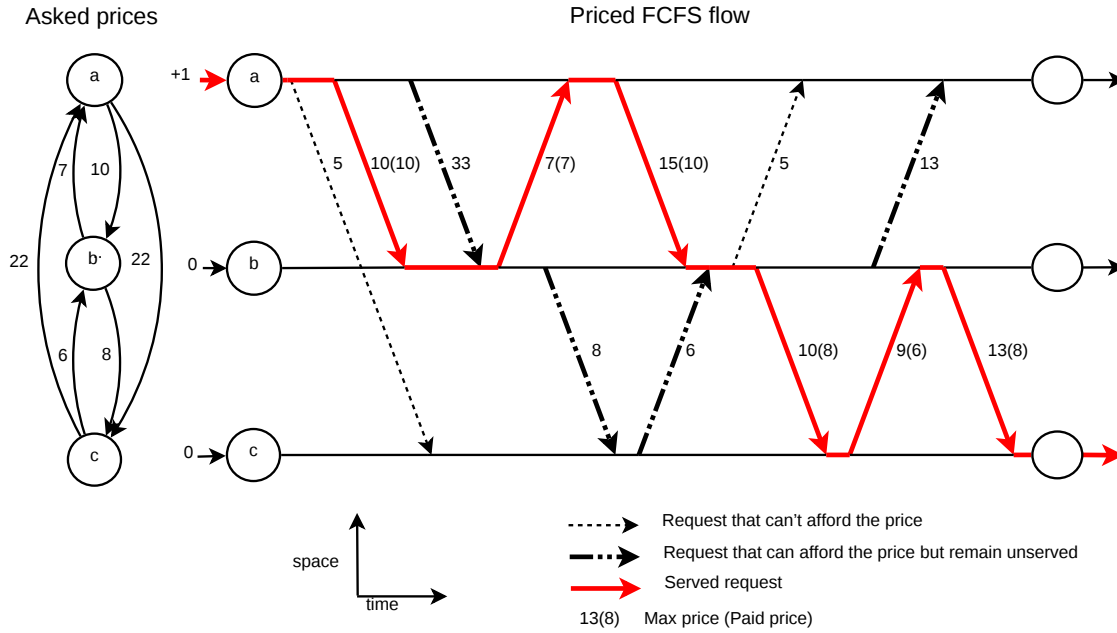


Figure 2: Priced First Come First Serve Flow with value 49.

pricing problem which sets, for each station, a price to take and a price to return a vehicle. Note that the complexity results can directly be extended to "time dependent prices", for instance prices that are allowed to be different in the morning, middle of day and evening (which might help to control tide phenomenon).

3.1 FCFS Flow Trip Pricing problem

We now define the MAX FCFS FLOW TRIP PRICING Problem which consists in setting a price for each trip in order to maximize the gain of the induced priced FCFS Flow.

Problem 1 MAX FCFS FLOW TRIP PRICING

- **INSTANCE:** *A set of stations S with capacities K_s for $s \in S$, a number N of vehicles, the vehicles distribution among the stations at the beginning of the horizon, a set $R = \{(s_o^r, t_o^r, s_d^r, t_d^r, p_{\max}^r), r \in R\}$ of trip requests.*
- **SOLUTION:** *The prices $p : S^2 \rightarrow \mathbb{R}$ to take a trip.*
- **MEASURE:** *The generated gain induced by the priced FCFS flow with prices p .*

We have the following complexity results.

Theorem 1 MAX FCFS FLOW TRIP PRICING problem is NP-hard, even with one vehicle, infinite station capacities and a unique maximum price.

Proof: STILL DRAFT

We show that we can solve any instance (with n variables and m clauses) of the NP-hard problem 3-SAT (Garey and Johnson, 1979) with an instance of the MAX FCFS FLOW TRIP PRICING problem.

We build a system that has one vehicle, and stations of infinite capacities. Figure 3 schemes an example of such reduction with two clauses. For each variable l , we define one station when it is unassigned: $\dot{l}$; and two when it is assigned: l and its complement $\bar{l}$. We define also two special stations res and tmp . There is only one vehicle in the system and it is located at station res at the beginning of the horizon.

We create the time and space demands with unique maximum price 1 as follows: We take iteratively each clause $a \vee b \vee c$ containing variables $\dot{a}$, $\dot{b}$ and $\dot{c}$. At time 1 there is a demand from station res to the station representing the first variable: $\dot{a}$. At time 2 we model the assignment of variable $\dot{a}$ with two demands: from stations $\dot{a}$ to a and then from $\dot{a}$ to $\bar{a}$. At time 3 we add a demand from the station representing the literal contained in the clause, a , to station res . Then we add another demand from station $\bar{a}$ representing the complement of literal contained in the clause to the station representing the next variable $\dot{b}$. At time 4 we have two successive demands,

from stations res to tmp and then from station tmp to res . At time 5 we construct the same series of demand as in times 2,3 and 4 but with the next variable $\dot{b}$. At time 6 we finally treat the last variable, $\dot{c}$ by assigning it to a literal in the same way as in time 2 and then by adding a demand from the station representing the literal c contained in the clause, to station res . This construction is then repeated for the next clause.

For a given clause, in the associated demands the longest weighted path has a length 7 and gain 7. There is 3 different longest weighted path but all are starting and ending at station res . For a given clause the maximum possible gain is then 7 and it is reached if and only if the assignment of variables satisfy the current clause.

Finally 3-SAT has a solution if and only if the instance of MAX FCFS FLOW TRIP PRICING has a solution with gain $7m$ which proves that MAX FCFS FLOW TRIP PRICING is NP-hard.

□

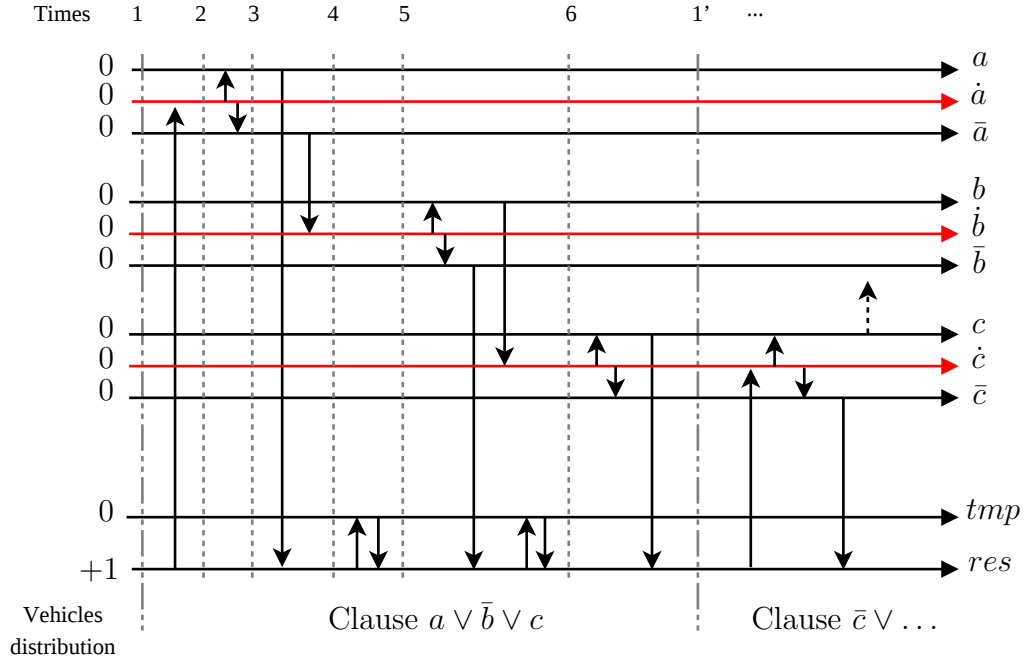


Figure 3: Reduction of 3-SAT to FCFS FLOW TRIP PRICING. Example with clauses $(a \vee \bar{b} \vee c) \wedge (\bar{c} \vee \dots)$.

3.2 FCFS Flow Station Pricing problem

We are now considering another way to set the prices $p(a, b)$ to take a trip $(a, b) \in S^2$. It is an aggregation (addition) of a price $p_t(a)$ to take a vehicle in station a and $p_r(b)$ to return it in station b : $p(a, b) = p_t(a) + p_r(b)$. We name it the MAX FCFS FLOW STATION PRICING Problem.

This type of pricing has an interest in a context where users have several possibilities for

origin/destination stations. It can help them to figure out quickly the different options they have to take a trip. For example with the use of price heated maps as in Papanikolaou (2011): Stations are coloured depending on their prices, for instance from yellow for cheap to red for expensive.

In this paper we are interested in studying the complexity of MAX FCFS FLOW STATION PRICING. We show that this problem is already NP-hard in the single choice context, that is only users have one possibility for origin/destination.

A REPHRASER: Moreover without loss of generality we consider that prices are independent from the distance/time the vehicle is used.

Problem 2 MAX FCFS FLOW STATION PRICING

- **INSTANCE:** *A set of stations S with capacities K_s for $s \in S$, a number N of vehicles, the vehicles distribution among the stations at the beginning of the horizon, a set $R = \{(s_o^r, t_o^r, s_d^r, t_d^r, p_{\max}^r), r \in R\}$ of trip requests.*
- **SOLUTION:** *Prices to take and return a vehicle at a station, p_t and p_r : $S \rightarrow \mathbb{R}$.*
- **MEASURE:** *The generated gain induced by the priced FCFS flow with prices $p(a, b) = p_t(a) + p_r(b)$.*

Theorem 2 MAX FCFS FLOW STATION PRICING is NP-hard even with one vehicle and infinite station capacities.

Proof: We can solve a MAX FCFS FLOW TRIP PRICING instance with a MAX FCFS FLOW STATION PRICING instance with the following construction: We use the same set of stations plus for each possible trip (a, b) we create two new stations, $ab1$ and $ab2$. Then for each trip request $r = (s_o^r = a, t_o^r, s_d^r = b, t_d^r, p_{\max}^r)$ we create three trip requests: $(a, t_o^r, ab1, t_o^r + \epsilon, 0)$, $(ab1, t_o^r + 2\epsilon, ab2, t_o^r + 3\epsilon, p_{\max}^r)$ and $(ab2, t_o^r + 4\epsilon, b, t_d^r, 0)$, with ϵ such that $4\epsilon < t_d^r - t_o^r$. Finally MAX FCFS FLOW TRIP PRICING has a solution of gain at least g if and only if the built instance of MAX FCFS FLOW STATION PRICING has a solution of gain at least g .

We know from the previous section that MAX FCFS FLOW TRIP PRICING is NP-hard even with one vehicle, therefore MAX FCFS FLOW STATION PRICING is also NP-hard even with one vehicle and infinite station capacities. $\square$

The opposite reduction is not trivial. In fact there is another difficulty in the MAX FCFS FLOW STATION PRICING not linked with the flow constraint: the quadratic prices assignment.

To prove that MAX FCFS FLOW STATION PRICING is NP-hard even without the flow constraint we define two new NP-hard graph optimization problems: MAX ORIENTED VERTEX PRICING and its unoriented version MAX VERTEX PRICING. They are both simplifications of MAX FCFS FLOW STATION PRICING.

Let $G(V, A, c)$ be a weighted directed multi-graph. Vertices V represent the stations and arcs $a \in A$ the trip requests with a weight c_a for its maximum price. The problem is to set two prices to take and return a vehicle, $p_t(i)$ and $p_r(i)$, for each vertex $i \in V$ in order to maximize the total gain on the arcs. A gain of $p_t(i) + p_r(j)$ is generated for each arc $(i, j) \in A$ if and only if $p_t(i) + p_r(j) \leq c_{i,j}$. More formally:

Problem 3 MAX ORIENTED VERTEX PRICING

- INSTANCE: *A weighted directed multi-graph $G(V, A, c)$ with $c : A \rightarrow \mathbb{R}$.*
- SOLUTION: *Prices p_t and $p_r : V \rightarrow \mathbb{R}$.*
- MEASURE: *The generated gain:*

$$\sum_{\substack{(i,j) \in A \\ p_t(i) + p_r(j) \leq c_{i,j}}} p_t(i) + p_r(j).$$

We extend the previous definition to weighted undirected multi-graph $G(V, E, c)$. We have now to set only one price $p(i)$ for each vertex $i \in V$ in order to maximize the total gain on the edges. A gain of $p(i) + p(j)$ is generated for each edge $(i, j) \in E$ if and only if $p(i) + p(j) \leq c_{i,j}$. More formally:

Problem 4 MAX VERTEX PRICING

- INSTANCE: *A weighted undirected multi-graph $G(V, E, c)$ with $c : E \rightarrow \mathbb{R}$.*
- SOLUTION: *Prices $p : V \rightarrow \mathbb{R}$.*
- MEASURE: *The generated gain:*

$$\sum_{\substack{(i,j) \in E \\ p(i) + p(j) \leq c_{i,j}}} p(i) + p(j).$$

To study the complexity of the last two problems, first recall the MAXIMUM NOT-ALL-EQUAL 3-SATISFIABILITY problem (NAE-3-SAT) which is known to be NP-hard, see Garey and Johnson (1979):

Problem 5 MAXIMUM NOT-ALL-EQUAL 3-SATISFIABILITY

- INSTANCE: *A set U of n variables and a collection C of m disjunctive clauses of 3 literals, where a literal is a variable or a negated variable in U .*

- **SOLUTION:** A truth assignment for U and a subset $C' \subseteq C$ of the clauses such that each clause in C' has at least one true literal and at least one false literal.
- **MEASURE:** $|C'|$.

We now present a polynomial reduction from NAE-3-SAT to MAX VERTEX PRICING to show that the latter is NP-hard.

Theorem 3 MAX VERTEX PRICING is NP-hard even on bipartite graphs.

Proof: We show a polynomial transformation from any NAE-3-SAT instances to a MAX VERTEX PRICING instance defining a bipartite graph. Figure 4 schemes it on an example.

Consider a general NAE-3-SAT instance with n variables and m clauses. Each variable l generates a q multi-edges $C4$ with vertices $\{l, l', \bar{l}, \bar{l}'\}$. q will be defined later on. The q edges between (l, l') and $(\bar{l}, \bar{l}')$ have weight 1, the q edges between $(l, \bar{l}')$ and $(l', \bar{l})$ have weight A (with A any constant ≥ 4). Each clause $(a \vee b \vee c)$ generates three edges of weight $A + 1$ between the vertices representing its literals: (a, b') , (b, c') and (c, a') . Note that this construction forms a bipartite graph since there are only edges between “normal” vertices and “prime” vertices.

The optimal solution for a multi-edged $C4$ representing variable l is worth $q \times (2A + 1)$. As scheme Figure 5, it is $p_l = A - \epsilon$, $p_{l'} = A - 1 + \epsilon$, $p_{\bar{l}} = 1 - \epsilon$ and $p_{\bar{l}'} = \epsilon$ with $\epsilon \in [0, 1]$ up to an exchange of $l \leftrightarrow \bar{l}$ and $l' \leftrightarrow \bar{l}'$. Note that all other pricing would at least make a loss of q . We say that a variable l is true (t) when $p_l \geq 3$, $p_{l'} \geq 3$ and when $p_{\bar{l}} \leq 1$, $p_{\bar{l}'} \leq 1$. We say that it is false (f) when $p_l \leq 1$ and $p_{l'} \leq 1$ and when $p_{\bar{l}} \geq 3$ and $p_{\bar{l}'} \geq 3$.

We set q such that optimizing our system iteratively starting by the multi-edged $C4$ (assigning the variables) and then dealing with the clauses is optimal. It ensures that each literal is either true (t) or false (f). There are only 4 possible types of clause generating different gains: $(f \vee f \vee f) = 3$, $(f \vee f \vee t) = 2A + 1$, $(f \vee t \vee t) \in [2A - 1, 2A + 1]$ and $(t \vee t \vee t) = 0$. We deduce that m clauses generates at most a gain of $m \times (2A + 1)$ and to have an ordered optimization optimal we only need to set $q > m \times (2A + 1)$.

In the end there exists a satisfiable assignment to NAE-3SAT if and only if there exists a solution for the corresponding instance of MAX VERTEX PRICING that is worth at least $q \times n \times (2A + 1) + m \times (2A - 1)$. It proves that MAX VERTEX PRICING is NP-hard. $\square$

We now present a polynomial reduction from MAX VERTEX PRICING to MAX ORIENTED VERTEX PRICING to show the latter is NP-hard.

Theorem 4 MAX ORIENTED VERTEX PRICING is NP-hard even on bipartite graphs.

Proof: From theorem 3 we know that MAX VERTEX PRICING is NP-hard even on bipartite graphs. Moreover a bipartite graph $G(V_1, V_2, E)$ can be oriented such that all vertices of V_1 are sources and all vertices of V_2 are sinks. With such construction we can solve MAX VERTEX

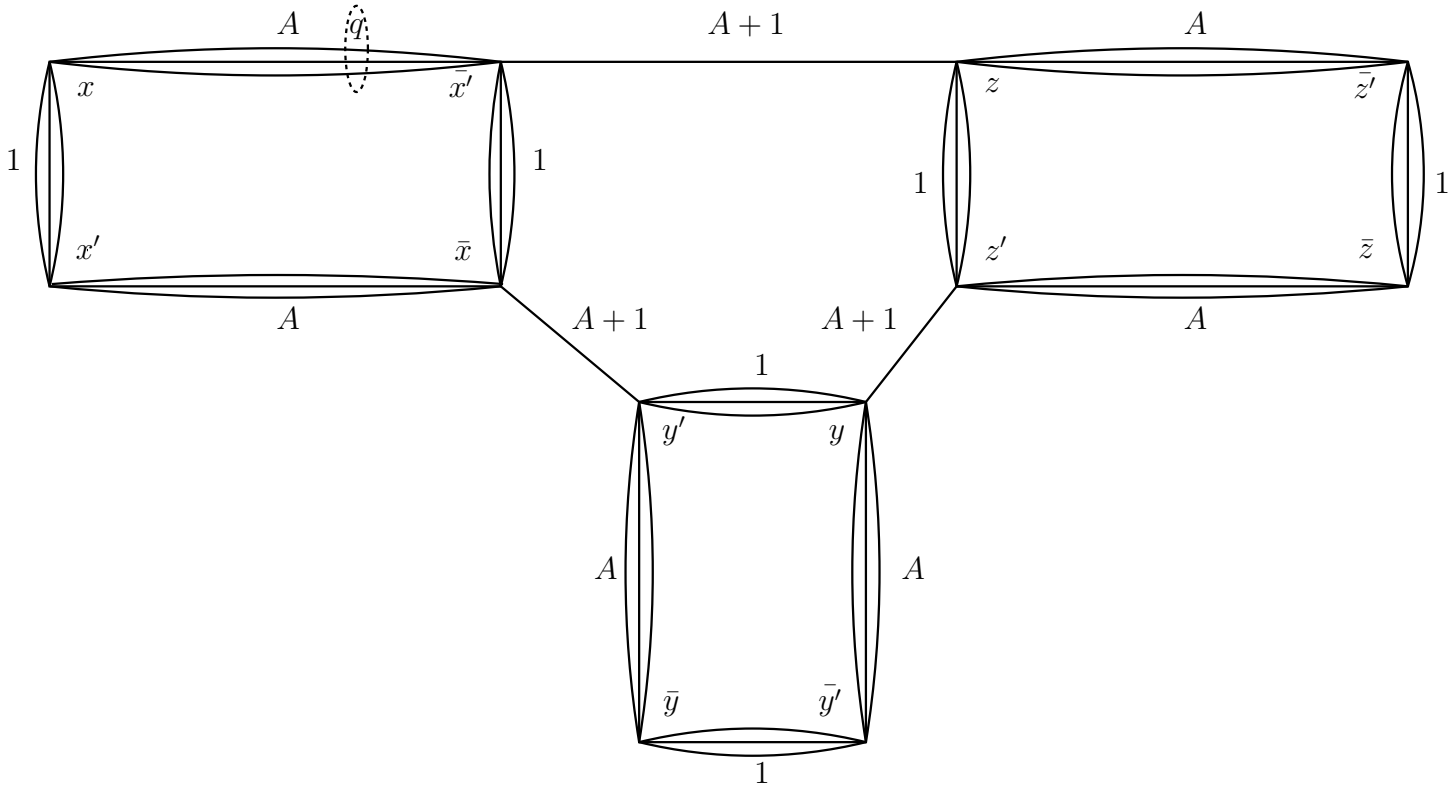


Figure 4: Reduction of NAE-3-SAT to MAX VERTEX PRICING on a bipartite graph. Example with clause $\bar{x} \vee y \vee z$.

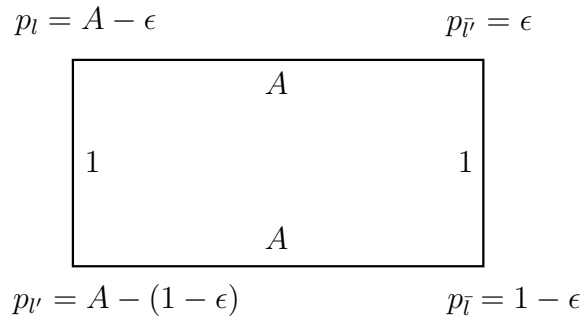


Figure 5: Optimal pricing for a C_4 representing a variable $l = \text{true}$.

PRICING on a bipartite graph with an instance of MAX ORIENTED VERTEX PRICING. Hence MAX ORIENTED VERTEX PRICING is NP-hard even on bipartite graph. $\square$

We use now the fact that MAX ORIENTED VERTEX PRICING is NP-hard to return to our original problem, MAX FCFS FLOW STATION PRICING and to show its complexity.

Theorem 5 MAX FCFS FLOW STATION PRICING is NP-hard, even with an unlimited number of vehicles, infinite station capacities and requests defining a bipartite graph.

Proof: Solving an instance of MAX FCFS FLOW STATION PRICING with an unlimited number of vehicles and infinite station capacities is equivalent to solve an instance of MAX ORIENTED VERTEX PRICING taking each request as an arc. MAX ORIENTED VERTEX PRICING is shown NP-hard on bipartite graphs, therefore MAX FCFS FLOW STATION PRICING is NP-hard even with requests defining a bipartite graph. $\square$

4 Station capacity problem

In the previous section we discussed the complexity of an operational problem (using pricing as leverage), which is based on the priced FCFS flow. In this section we study the complexity of a tactical problem: Setting the optimal capacity for each station given a set of requests and a system evolution following a FCFS flow.

Intuitively, without any additional constraints, one would like to set the station capacities $\mathcal{K}$ equal to the number of vehicles, *i.e.* $\forall s \in S, \mathcal{K}_s = N$. However it might be interesting to set smaller values for $\mathcal{K}$ in order to control the location of vehicles in the system (subject to tides phenomenon for instance). Station capacities are then used as a balancing tool. Figure 6 gives an example of station capacity optimization. For this instance the optimal capacity for station b is $\mathcal{K}_b = N/2$ while station a and c have a capacity $\geq N$. It will allow $N/2$ vehicles to take half of the trips from station a to station b at price 1 until station b is full. Then the remaining vehicles will have to wait in station b the requests going to station c at price 2. This policy generates the optimal a final profit of $3N/2$.

Problem 6 MAX FCFS FLOW STATION CAPACITIES

- **INSTANCE:** A set of stations S , a number N of vehicles with their repartition among the stations at the beginning of the horizon, a set of trip requests $r \in R$ to go from an origin station s_o^r at time t_o^r to a destination station s_d^r at time t_d^r for a price p^r .
- **SOLUTION:** A function $\mathcal{K} : S \rightarrow \mathbb{N}^+$ defining the capacity of each station.
- **MEASURE:** The gain generated by the FCFS flow with station capacities $\mathcal{K}$.

Once again this problem involving a FCFS flow is NP-hard.

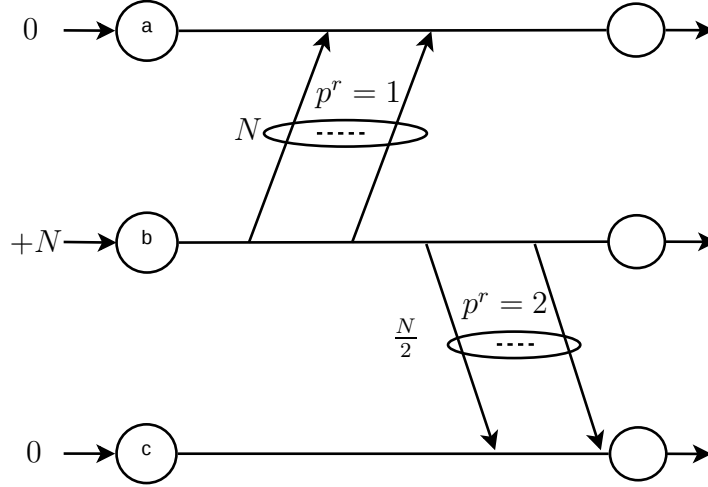


Figure 6: Example where proper station capacities increase the number of trips sold. Here $K_a = N/2$ and $K_b, K_c \geq N$ with a optimal profit of $3N/2$.

Theorem 6 MAX FCFS FLOW STATION CAPACITIES is NP-hard.

Proof: Let's show a reduction of the classical NP-hard problem 3-SAT (Garey and Johnson, 1979) to our capacity problem. Figure 7 schemes such reduction.

Let's consider m clauses and n variables. We have $2n + m + 2$ stations: two for each variable (for a true assignment of the variable l and its complement $\bar{l}$), one for each clause, one for the source s and one for the result res .

At the beginning of the horizon we have $3m$ vehicles available at station s , and none elsewhere. For each clause $c = l_1 \vee l_2 \vee l_3$ composed with variables v_1, v_2 and v_3 , we have a series of instants: At time 1, there are 3 requests of price 1 from the station source s to stations v_1, v_2 and v_3 . At time 2, there are 3 requests of price 1 from the station v_1, v_2 and v_3 to respectively station $\bar{v}_1, \bar{v}_2$ and $\bar{v}_3$. At time 3, there are 3 requests of price 1 from the stations l_1, l_2 and l_3 to station c . At time 4, we empty each station with 6 requests of price 1 from stations $v_1, v_2, v_3, \bar{v}_1, \bar{v}_2$ and $\bar{v}_3$ to station res . At time 5, there is one request of price $9m$ from the station c to station res . We repeat the same construction for the next clauses after time 5.

Optimal solutions have the following structure: For each variable l , station l has a capacity ≥ 1 . If variable l is assigned to true, station $\bar{l}$ has a capacity equals to 0. Otherwise if it assigned to false station $\bar{l}$ has a capacity ≥ 1 . Stations representing clauses have a capacity equal to 1. Station s and res have a capacity $\geq 3m$.

Finally 3-SAT is satisfiable if and only if this instance has a solution of gain at least $m \times (9m + 6)$, which shows that MAX FCFS FLOW STATION CAPACITIES is NP-hard. $\square$

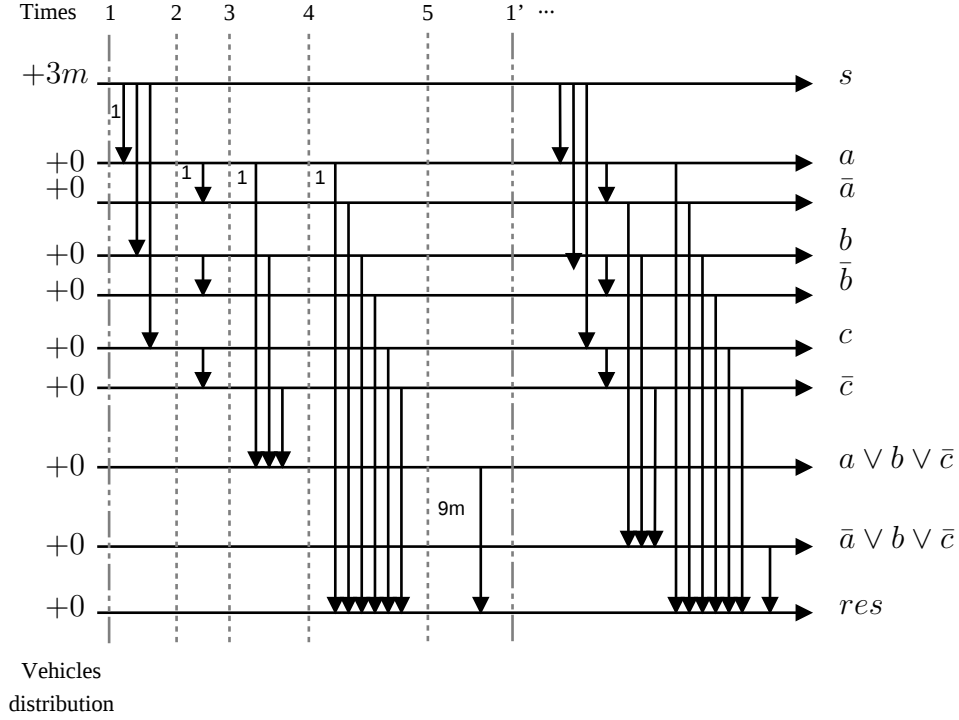


Figure 7: Reduction of 3-SAT to FCFS FLOW STATION CAPACITIES with clauses: $(a \vee b \vee \bar{c}) \wedge (\bar{a} \vee b \vee \bar{c})$.

5 Max Flow Upper Bound

Given that the previous FCFS flow problems are NP-hard, bounds or approximation algorithms might be of interest. A “classic” flow is a relaxation of the First Come First Serve flow evaluation. One of the most famous optimization problem on classic flows is the MAX FLOW problem which is polynomially solvable. MAX FLOW gives trivially an Upper Bound (UB) on many FCFS optimization problems (MAX FCFS FLOW TRIP PRICING...). However, as we are going to show in this section, in a general context there is no guaranty of quality for this bound and just at most a poor one for a subclass of problem.

Indeed, because MAX FLOW doesn’t take into account reservations of parking spot at the destination station it can give an UB arbitrarily far from the optimal value. Figure 8 gives an example with 2 stations of capacity 1 and 2 vehicles with q crossed demands. In this instance MAX FLOW is able to serve all q requests while any FCFS flow can’t serve any.

Now even if we consider unlimited station capacities, which means that we don’t have the problem of parking spot reservation anymore, MAX FLOW UB can again be arbitrarily far from the optimal value. Figure 9 schemes an example, with 2 stations Lower (L) and Upper (U) with 1 vehicle available at L at the beginning of the horizon and requests of maximum price 1. The first request goes from L to U and takes the entire horizon to reach the station U. Then there is q successive trip requests from L to U and from U to L. In this instance MAX FLOW is able

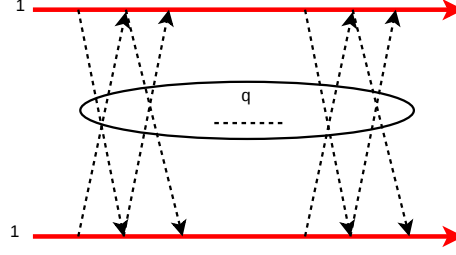


Figure 8: MAX FLOW UB doesn't consider parking spot reservation.

to serve q requests rejecting only the first long one while any FCFS flow policy can't serve more than one, the first one.

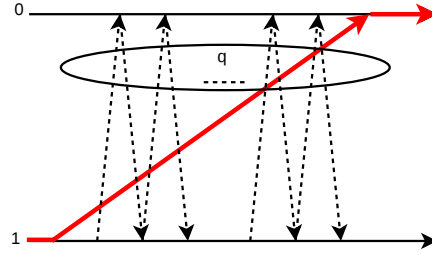


Figure 9: MAX FLOW is an unbounded UB even with infinite station capacities.

The previous example used different transportation times for the same trip. Nevertheless, with unitary transportation times and still infinite station capacity MAX FLOW UB can still be $\frac{M \times (M-1)}{2}$ times better than any feasible FCFS solution (where M is the number of stations). Figure 10 schemes how to create the family of instance reaching this bound on an example having 3 stations Lower (L), Center (C) and Upper (U) with 1 vehicle available at C and requests of maximum price 1. At time 0 we have a first trip request from C to L followed by q successive trip requests from C to U and from U to C. At time 1 we have q successive trip requests from U to L and from L to U. At time 2 we have finally q successive trip requests from L to C and from C to L. In this instance MAX FLOW can serve $3 \times q - 2$ requests rejecting only the first trip request from C to L and the last request from U to C and from L to U, while any FCFS flow policy can serve at most $q + 1$ requests choosing to serve the q requests from C to L, from C to L or from L to U. This example can be extended to M stations to obtain when q tends to infinity the $\left(\frac{M \times (M-1)}{2}\right)$ upper bound.

We observe in practice that MAX FLOW UB is much closer to the optimal FCFS solution than the poor bound $\left(\frac{M \times (M-1)}{2}\right)$ factor so it still could be useful.

One could think to use MAX FLOW solution to build an approximation algorithm. For instance for each trip, set the minimum price of the served requests, or close it, if there is none. However, with the same arguments used for the MAX FLOW UB, this heuristic would give only at least $\left(\frac{M \times (M-1)}{2}\right)$ - approximation.

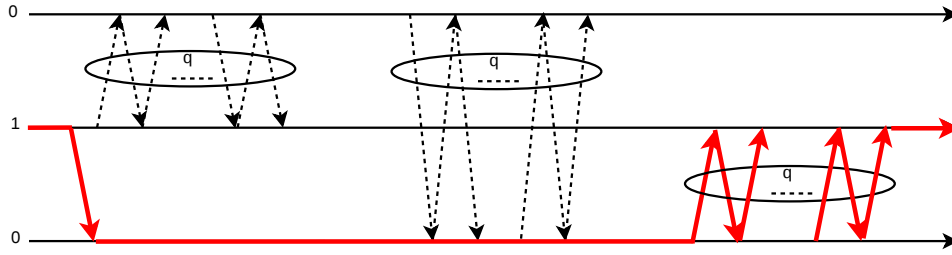


Figure 10: MAX FLOW UB can be $\frac{M \times (M-1)}{2}$ times better than any FCFS solution even with infinite station capacity and unitary transportation times.

References

- Autolib'. <http://www.autolib-paris.fr>.
- Bixi. <http://montreal.bixi.com>.
- Budget Truck Rental. <http://www.budgettruck.com/>.
- Car2go. <http://www.car2go.com>.
- W.J. Carroll and R.C. Grimes. Evolutionary change in product management: Experiences in the car rental industry. *Interfaces*, 25(5):84104, 1995.
- D. Chemla, F. Meunier, and R. Wolfler-Calvo. Balancing the stations of a self-service bike hiring system. 2011.
- C. Contardo, C. Morency, and L-M. Rousseau. Balancing a dynamic public bike-sharing system. Technical Report 09, CIRRELT, 2012.
- P. DeMaio. Bike-sharing: History, impacts, models of provision, and future. *Journal of Public Transportation*, 12(4):41–56, 2009.
- C. Fricker and N. Gast. Incentives and regulations in bike-sharing systems with stations of finite capacity. *arXiv :1201.1178v1*, January 2012.
- C. Fricker, N. Gast, and H. Mohamed. Mean field analysis for inhomogeneous bike sharing systems. In *23rd Intern. Meeting on Probabilistic, Combinatorial, and Asymptotic Methods for the Analysis of Algorithms (AofA'12)*, 2012.
- M. R. Garey and D. S. Johnson. *Computers and intractability : a guide to the theory of NP-completeness*. A Series of books in the mathematical sciences. W. H. Freeman, San Francisco, 1979.
- 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(1):198 – 207, 2011. ISSN 0377-2217. doi: DOI:10.1016/j.ejor.2010.12.015. URL <http://www.sciencedirect.com/science/article/pii/S0377221710008817>.

- M. K. Geraghty and E. Johnson. Revenue management saves national car rental. *Interfaces*, 27 (1997 vol. 27 no. 1):107–127, January/February 1997.
- F. Guerriero, G. Miglionico, and F. Olivito. Revenue management policies for the truck rental industry. *Transportation Research Part E: Logistics and Transportation Review*, 48(1):202 – 214, 2012. ISSN 1366-5545. doi: 10.1016/j.tre.2011.07.006. URL <http://www.sciencedirect.com/science/article/pii/S1366554511001013>.
- A. Haensela, M. Mederer, and H. Schmidt. Revenue management in the car rental industry: A stochastic programming approach. 2011.
- J.R. Lin and Y. Ta-Hui. Strategic design of public bicycle sharing systems with service level constraints. *Transportation Research Part E: Logistics and Transportation Review*, 47(2):284 – 294, 2011. ISSN 1366-5545. doi: 10.1016/j.tre.2010.09.004. URL <http://www.sciencedirect.com/science/article/pii/S1366554510000839>.
- C. Morency, M. Trépanier, and F. Godefroy. Insight into the montreal bikesharing system. In *TRB-Transportation Research Board Annual Meeting, Washington, USA*, pages Paper #11–1238, 17 pages, January 2011.
- D. Papanikolaou. The market economy of trips. Master’s thesis, Massachusetts Institute of Technology, 2011.
- Rentn’Drop. <http://www.rentanddrop.com/>.
- D.C. Shoup. *The High Cost of Free-Parking*. Planners Press, Chicago, 2005.
- J. Shu, M. Chou, O. Liu, C.P. Teo, and I-L Wang. Bicycle-sharing system: Deployment, utilization and the value of re-distribution. , 2010. URL <http://bschool.nus.edu/Staff/bizteocp/BS2010.pdf>.
- Vélib’. <http://www.velib.paris.fr>.
- A. Waserhole, V. Jost, and N. Brauner. Vehicle sharing system pricing regulation: A fluid approximation. <http://hal.archives-ouvertes.fr/hal-00727041>, 2012. URL <http://hal.archives-ouvertes.fr/hal-00727041>.