



HAL
open science

Vehicle Sharing System Pricing Regulation: Deterministic Approach, Complexity results

Ariel Waserhole, Vincent Jost, Nadia Brauner

► **To cite this version:**

Ariel Waserhole, Vincent Jost, Nadia Brauner. Vehicle Sharing System Pricing Regulation: Deterministic Approach, Complexity results. 2012. hal-00727040v1

HAL Id: hal-00727040

<https://hal.science/hal-00727040v1>

Preprint submitted on 31 Aug 2012 (v1), 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

August 31, 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 approximation on FCFS flow problems.

1 Introduction

1.1 Context

Shoup [14] reports that, based on a sample of 22 US studies, cars looking for a parking spot contribute to 30% of the city traffic. Moreover most cars are not in use 90% of the time (use 2 hours per day) but still occupying parking spots! Could we have less vehicles for the same demand satisfaction?

Recently, the interest in Vehicle Sharing Systems (VSS) in cities has increased significantly. Indeed nowadays, 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 traffic congestion (parking issues), noise, and air pollution (using 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 [7]. 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 Sharing System (BSS) Vélib' [4]. It was implemented in Paris in 2007 and now has more than 1200 stations and 20 000 vehicles selling around 110 000 trips per day. It has inspired several other cities all around the world, and now more than 300 cities have such a system, including Montréal, Beijing, Barcelona, Mexico City, Tel Aviv [7].

1.2 One-way Vehicle Sharing Systems: a management issue

However if the 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 because vehicles aren't the only key resources 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 BSS, problems of bikes and parking spots availability have appeared very often. Causes are various but we can highlight two important phenomenons: 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 have been studied. At a strategic level, some authors considered the optimal capacity and locations of bike rental stations. Shu et al. [15] proposed a stochastic network flow model to support these decisions. They used their model to design a BSS in Singapore based on demand forecast derived from current usage of the mass transit system. Lin and Ta-Hui [12] considered a similar problem but formulated 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 have investigate the optimal number of vehicles given a set of station. George and Xia [11] have studied the fleet sizing problem with constant demand and no parking capacity. Fricker and Gast [8, 9] looked 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 showed 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 $\frac{1}{\mathcal{K}+1}$ that any given station is empty or full.

At an operational level, when the system is not able to meet the demand with a reasonable standard of quality, currently in BSS, 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 in a full station. In the literature many papers deal already with this problem. A static version of the BSS balancing problem is treated in Chemla et al. [5] and a dynamic one in Contardo et al. [6].

1.3 Our scope: Self Regulation Systems

A new type of VSS have appeared lately: one-way Car Sharing Systems (CSS) with Autolib' in Paris [2] and Car2go [3] in more than 10 cities (Vancouver, San Diego, Lyon, Ulm...). With cars, operational balancing optimization through relocation seems inappropriate due to their size. This is the reason why we are looking here into a different operational management approach: Self Regulating Systems through pricing. In this study, assuming that demand is elastic, we use the prices to influence user choices and drive the system towards its most efficient dynamic.

A model for the Stochastic Vehicle Sharing System Pricing problem have been proposed in Wasserhole and Jost [16]. They model it as a Markov Decision Process, shown intractable, and therefore 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. Indeed it is a classic and natural way to study deterministic version of stochastic problems. Here we consider that all trip requests are available at the beginning of the time horizon. It is like optimizing a posteriori the system, using exploitation data (scenario of the day), and it might give some informations about the system.

On the one hand, such optimization gives an Upper Bound of the on line stochastic optimization problem on a given instance. On the other hand, solving efficiently the deterministic problem on a scenario allows us to do some robust optimization heuristics: By optimizing on a set of possible scenarii generated with a demand forecast we can expect to derive some results that can be applied on-line for the stochastic problem. Moreover Morency et al. [13] showed that in Bixi Montreal [1], 68% of the trips were made by members and that their frequencies of use are quite stable across the week. We can hence consider that this demand is quasi deterministic and considering deterministic requests might be a good approximation.

In the following we are going to focus on complexity results on the deterministic Vehicle Sharing System Pricing problem and show that the combinatorial problems involved are NP -hard.

2 First Come First Serve flow constraint

Let's first define a new type of flow constraint, the "First Come First Serve flow" (FCFS flow) over a time and space network. Even if it was not explicitly specified (named) we can consider that this constraint is already known in continuous time models. Indeed it arises naturally in many applications, for instance in the fluid approximation of a Markov Decision Process [16], however to the best of our knowledge it hasn't been studied, nor mentioned, in a discrete time model.

In order to stay stay in the Vehicle Sharing Systems lexical field we speak about a flow of vehicles transiting thanks to user trips along stations. In a more general context it could be called a flow of resources along locations.

2.1 First Come First Serve flow in Time and Space Network

Let's consider a deterministic process of a flow of N vehicles transiting among a set of stations S of capacity K_s , $s \in S$ on an horizon time $H = [0, T]$. At time 0 we have an initial repartition of the vehicles among the stations. The vehicles then transit between stations by accepting spatio-temporal on-line user trip requests. The set of trip requests R is known at the beginning of the horizon. A trip request $r \in R$ wants to go from an original station s_o^r at time t_o^r to a final station s_f^r at time t_f^r for a price p^r .

Given such data we can build a time and space network and follow in linear time 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 $r = (s_o^r, t_o^r = t, s_f^r, t_f^r, p^r) \in R$ 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_f^r 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_f^r until time $t_f^r \geq t$. At time $t = t_f^r$ the vehicle is finally available again at station s_f^r .

This deterministic process is called "First Come First Serve Flow" and the gain generated can be evaluated in linear time. Figure 1 gives an example of a First Come First Serve flow with three stations, 12 requests and one 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 flow can be evaluated in a linear time. Hence the decision versions of the optimization problems considered on this paper are in NP.*

2.2 Pricing in First Come First Serve flow

Let's now consider that the system has establish a fixed price $p_{a,b}$ for each trip $(a, b) \in S^2$. The price of each request is now to be interpreted as the maximum that the user is willing to pay for

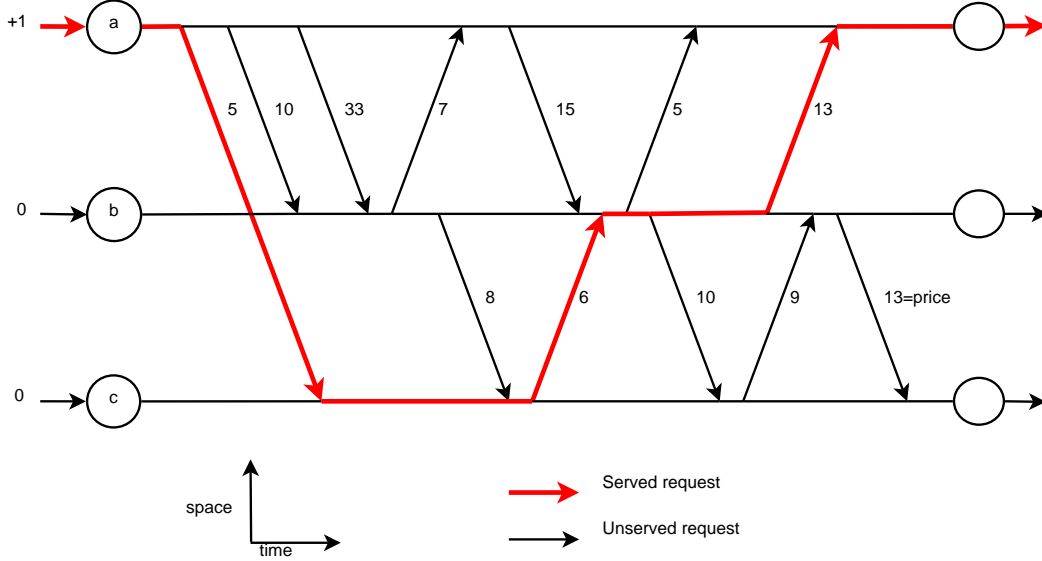


Figure 1: First Come First Serve Flow with value 24.

taking the trip. We denote it p_{\max}^r , $r \in R$ from now on. We have that only some requests can afford to pay the asked prices $R_p = \{r \in R : p_{\max}^r \geq p_{s_o^r, s_f^r}\}$. Finally when a request $r \in R$ to go from a station $a = s_o^r$ to a station $b = s_f^r$ is accepted, *i.e.* $p_{\max}^r \geq p_{a,b}$ and there is a vehicle and a parking spot available at time t_o^r for the trip (a, b) , it generates a gain $p_{a,b}$ (and not p_{\max}^r).

This deterministic process is called “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 price $p : S^2 \rightarrow \mathbb{R}$ and can also be evaluated in linear time.

Figure 2 gives an example of an evaluation of such process with three stations and one vehicle. Here, out of 12 requests, 10 can afford the asked price and finally 6 are served for a gain of 47.

3 Pricing problems

We are now defining optimization problems based on the Priced First Come First Flow defined in the previous section. We want to optimize a system through price leverage: We are looking into pricing policies to maximize the gain of the induced priced FCFS flow. These policies can be *Dynamic*, *i.e.* dependent on system’s state (vehicle repartition and period of the day), or *Static* *i.e.* independent on system’s state, set in advance but function of the trip and the time of the day.

In this paper we focus on static pricing, trip dependent but constant over the time horizon. We demonstrate the complexity of two optimization problems: The FCFS flow trip pricing problem and the FCFS flow station pricing problem are *NP*-hard. Note that the complexity results can directly be extended to prices that are time step dependent, for instance different in the morning, middle of day and evening to control tides phenomenon.

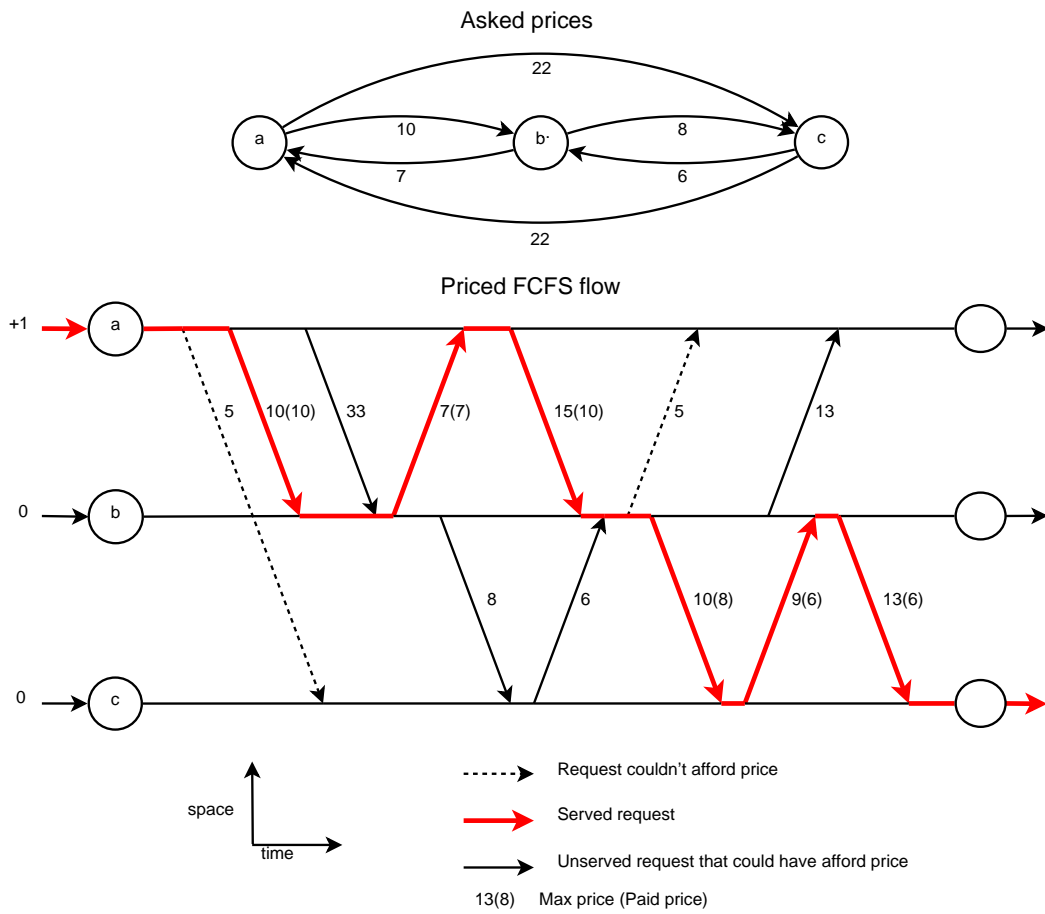


Figure 2: Priced in First Come First Serve Flow with value 47.

3.1 FCFS Flow Trip Pricing problem

Let's define the MAX FCFS FLOW TRIP PRICING Problem which consists in setting a price for each trip in order to maximize the induced Priced FCFS Flow.

Definition 1 (Max FCFS Flow Trip Pricing)

- **INSTANCE:** *A set of stations S with specified capacity, a number N of vehicles available with their repartition at the beginning of the horizon, a set of trip requests $r \in R$ to go from an original station s_o^r at time t_o^r to a final station s_f^r at time t_f^r with a maximum price p_{\max}^r .*
- **SOLUTION:** *A function $p : S^2 \rightarrow \mathbb{R}$ setting the price to take each trip.*
- **MEASURE:** *The generated gain induced by the priced FCFS flow with prices p , i.e. the FCFS flow with requests that could afford the asked price to take the trip.*

For static pricing we have the following complexity results.

Theorem 1 (Max FCFS Flow Trip Pricing complexity) *Static MAX FCFS FLOW TRIP PRICING is NP-hard, even with infinite station capacities.*

Proof : We show that we can solve any instance (with n literals and m clauses) of the NP-hard problem 3-SAT [10] with an instance of the MAX FCFS FLOW TRIP PRICING problem.

For each literal l we define two stations, one representing the literal active l and the other one representing it inactive \bar{l} . Station l has m vehicles available at the beginning of the horizon. Just after the horizon started, we have m successive requests of maximum price 1 to go from l to \bar{l} .

For each clause c we define a station c . At the middle of the horizon we have a request of maximum price 1 from each literal composing the clause c to the station c . At the very end of the horizon we have a request of maximum price $m(n+3)+1$ to go from station c to station res . Figure 3 schemes an example of such reduction with two clauses.

Finally 3-SAT has a solution if and only if the instance of MAX FCFS FLOW TRIP PRICING has a solution with gain at least $m \times (m(n+3)+1)$, which shows that MAX FCFS FLOW TRIP PRICING is NP-hard.

□

3.2 FCFS Flow Station Pricing problem

We are now considering another way to set the prices to take each trip $(a, b) \in S^2$. It is an aggregation (addition) of a price to take a vehicle in station a and to return it in b . This type

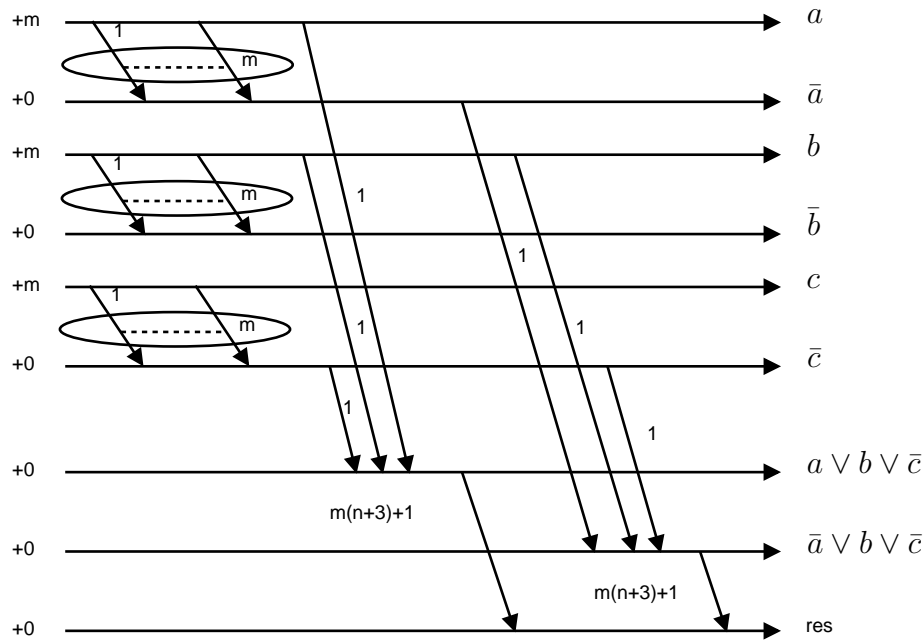


Figure 3: Reduction of 3-SAT to FCFS FLOW TRIP PRICING. Example with clauses $(a \vee b \vee \bar{c}) \wedge (\bar{a} \vee b \vee \bar{c})$.

of pricing has an interest for the user that can understand more easily the different options he has to take a trip since prices can be represented by colouring the stations. We call it the MAX FCFS FLOW STATION PRICING Problem. The context of users having several possibilities for origin/destination stations is not considered explicitly in this paper since it increases the complexity of the considered problems.

Definition 2 (Max FCFS Flow Station Pricing)

- **INSTANCE:** A set of stations S with specified capacity, a number N of vehicles available with their repartition at the beginning of the horizon, a set of trip requests $r \in R$ to go from an original station s_o^r at time t_o^r to a final station s_f^r at time t_f^r with a maximum price p_{\max}^r .
- **SOLUTION:** Two price functions to take or return a vehicle p_t and $p_r: S \rightarrow \mathbb{R}$.
- **MEASURE:** The generated gain induced by the priced FCFS flow with prices $p(a, b) = p_t(a) + p_r(b)$, i.e. the FCFS flow with requests that could afford the asked price to take the trip.

One could think that there is an easy reduction between MAX FCFS FLOW STATION PRICING and MAX FCFS FLOW TRIP PRICING, however, although these problems are really close we haven't be able to build a simple polynomial reduction linking them.

To prove that MAX FCFS FLOW STATION PRICING is *NP*-hard we are going to use a reduction to new *NP*-hard Graph Optimization problems defined in the following: MAX VERTEX PRICING and MAX ORIENTED VERTEX PRICING.

Definition 3 (Max Oriented Vertex Pricing) Let $G(V, A, c)$ be a weighted directed multi-graph, with vertices V (stations) and edges $a \in A$ (trip requests) with a weight c_a (request maximum price). The problem is to set a couple of prices (t_i, r_i) for each vertex $i \in V$ in order to maximize the total gain. A revenue of $t_i + r_j$ is generated for each arc $a = (i, j)$ if and only if $t_i + r_j \leq c_{i,j}$. More formally:

MAX ORIENTED VERTEX PRICING:

- INSTANCE: A weighted directed multi-graph $G(V, A, c)$ with $c : A \rightarrow \mathbb{R}$.
- SOLUTION: Two price functions t and $r : V \rightarrow \mathbb{R}$.
- MEASURE: The generated revenue:

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

Definition 4 (Max Vertex Pricing) Let $G(V, E, c)$ be a weighted undirected multi-graph, with vertices V (stations) and edges $e \in E$ (trip requests) with a weight c_e (request maximum price). The problem is to set a price p_i for each vertex $i \in V$ in order to maximize the total gain. A revenue of $p_i + p_j$ is generated for each edge (i, j) if and only if $p_i + p_j \leq c_{i,j}$. More formally:

MAX VERTEX PRICING:

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

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

To study the complexity of the two last problems, let's first recall the MAXIMUM NOT-ALL-EQUAL 3-SATISFIABILITY problem (NAE-3-SAT) which is known *NP*-hard, see Garey and Johnson [10]:

Definition 5 (Maximum Not-All-Equal 3-Satisfiability)

- INSTANCE: A set U of n variables and a collection C of m disjunctive clauses of at most 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 2 (Max Vertex Pricing complexity) MAX VERTEX PRICING is NP -hard even on bipartite graph.

Proof : We show how to create a polynomial transformation of NAE-3-SAT to an instance of the MAX VERTEX PRICING on a bipartite graph as schemed on an example figure 4.

Let's consider a general NAE-3-SAT instance with n literals and m clauses. Each literal l generates a $C4$ with vertices $\{l, l', \bar{l}, \bar{l}'\}$ and each of its four edges is replaced by a q multi-edges. The q edges between (l, l') and (\bar{l}, \bar{l}') are worth 1, the q edges between (l, \bar{l}') and (l', \bar{l}) are worth A (with A any constant ≥ 4). Each clause $(a \vee b \vee c)$ generates three edges of cost $A + 1$ between the vertices representing its literals: (a, b') , (b, c') and (c, a') . We can note that this construction forms a bipartite graph since there is only edges between "normal" vertices and "prime" vertices.

The optimal solution for a multi-edged $C4$ representing literal l is worth $2A + 1$ and is $l = A - \epsilon$, $l' = A - 1 + \epsilon$, $\bar{l} = 1 - \epsilon$ and $\bar{l}' = \epsilon$ with $\epsilon \in [0, 1]$ up to an exchange of $l \leftrightarrow \bar{l}$ and $l' \leftrightarrow \bar{l}'$. All other pricing would at least make a loss of q . We say that a literal l is true (t) when $l \approx l' \approx O(A)$ and $\bar{l} \approx \bar{l}' \approx O(1)$, and false (f) when $l \approx l' \approx O(1)$ and $\bar{l} \approx \bar{l}' \approx O(A)$.

We are setting q such that optimizing our system in a lexicographic order starting by the multi-edged $C4$ representing the literals 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 $m \times (2 \times A + 1)$ and to have a lexicographic optimization standing we only need to set $q > m \times (2 \times A + 1)$.

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 shows that MAX VERTEX PRICING is NP -hard. □

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

Theorem 3 (Max Oriented Vertex Pricing complexity) MAX ORIENTED VERTEX PRICING is NP -hard even on bipartite graph.

Proof : From theorem 2 we know that MAX VERTEX PRICING is NP -hard even on bipartite graph. 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 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. □

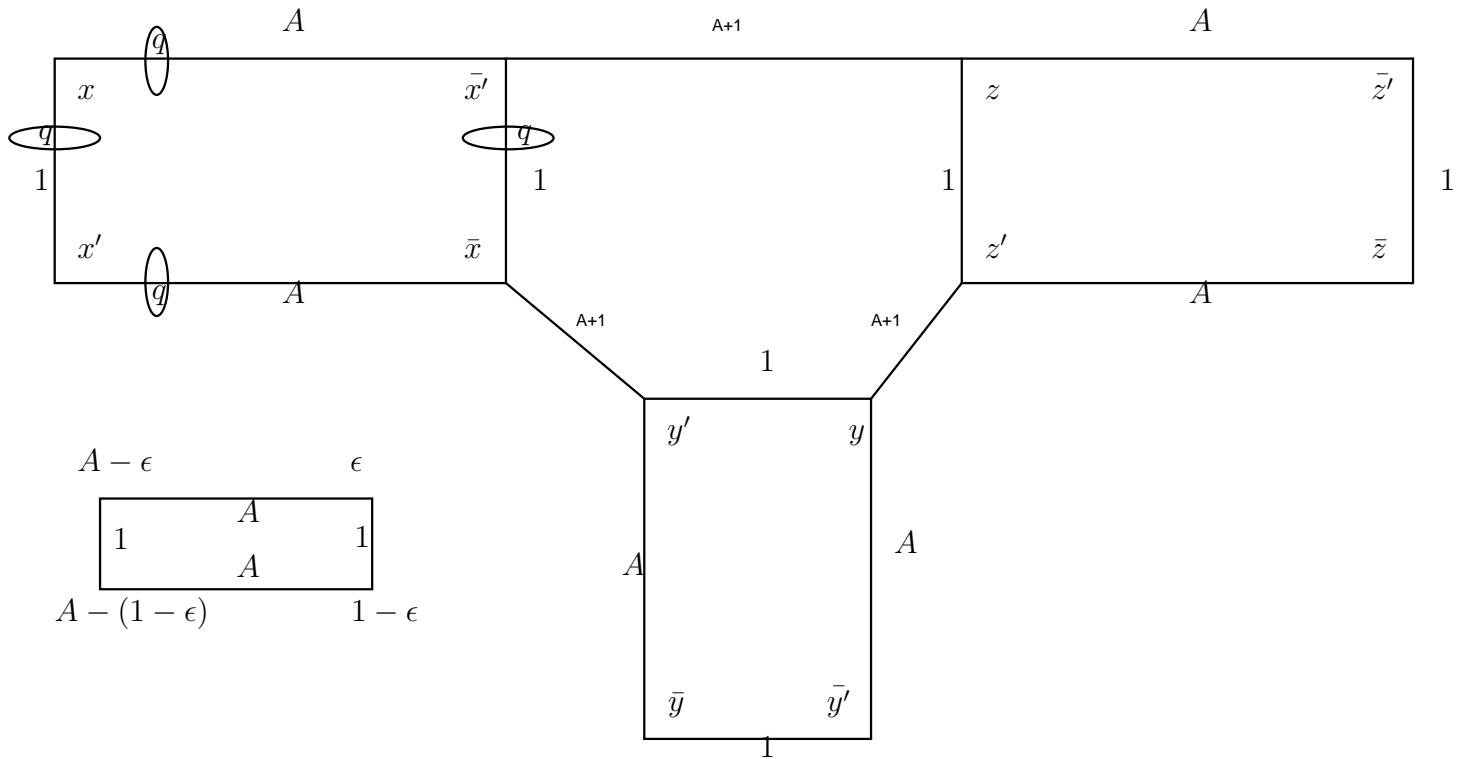


Figure 4: Reduction of NAE-3-SAT to MAX VERTEX PRICING on a bipartite graph. Example with clause $\bar{x} \vee y \vee z$. Bottom left figure shows an optimal pricing for a C_4 .

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

Theorem 4 (Max FCFS Flow Station Pricing complexity) MAX FCFS FLOW STATION PRICING is *NP*-hard, even with an unlimited number of vehicles and infinite station capacities (and constant requests defining a bipartite graph).

Proof : MAX FCFS FLOW STATION PRICING with an unlimited unlimited number of vehicles and infinite station capacities amounts in solving MAX ORIENTED VERTEX PRICING which is shown *NP*-hard on bipartite graph. \square

4 Station capacity problem

In the previous section we discussed about the complexity of an operational problem (using pricing as leverage), which was 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.

Definition 6 (Max FCFS Flow Station Capacities)

- **INSTANCE:** A set of stations S , a number N of vehicles available with their repartition at the beginning of the horizon, a set of trip requests $r \in R$ to go from an original station s_o^r at time t_o^r to a final station s_f^r at time t_f^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.

Theorem 5 MAX FCFS FLOW STATION CAPACITIES is *NP*-hard.

Proof : Let's show a reduction of the classical *NP*-hard problem [10] 3-SAT to our capacity problem. Figure 5 schemes such reduction.

Let's consider m clauses and n literals. We have $2n + m + 2$ stations: two for each state of each literal (l and \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 $3 \times m$ vehicles available at station s , and none elsewhere. For each clause we will first make 3 requests of price 1 going to the active part of the literal composing it. Then a request of price 1 between the active part of the literal and its complement. Then another request of price 1 from each literal of the clause c to station c . Then we empty each

station representing the two states of the literals composing the clause by adding a request of price 1 to station *res*. In the end we add a request of price $9m$ from station *c* to station *res* and continue with the next clause. Finally 3-SAT is satisfiable if and only if this instance have a solution of gain $m \times (9m + 9)$, which shows that MAX FCFS FLOW STATION CAPACITIES is NP-hard. \square

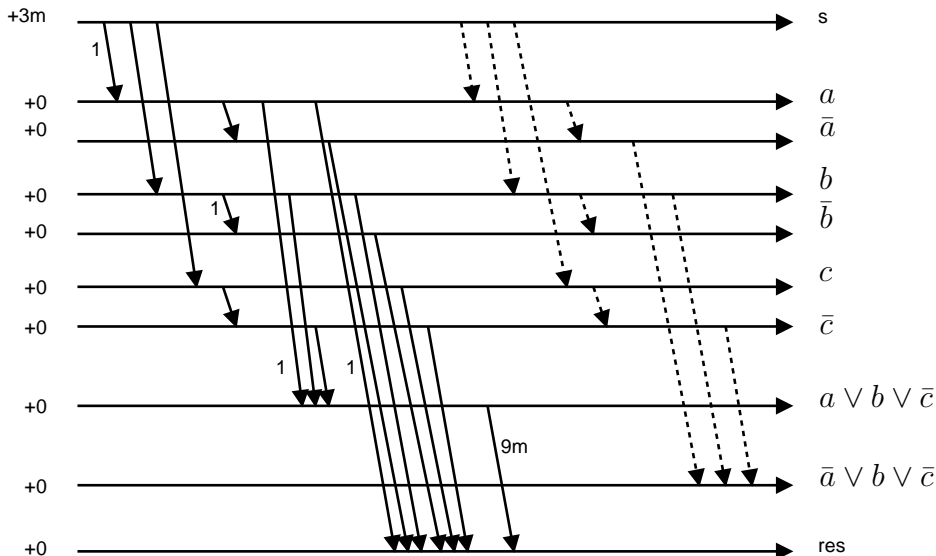


Figure 5: Reduction of 3-SAT to FCFS FLOW STATION CAPACITIES with clauses: $(a \vee b \vee \bar{c}) \wedge (\bar{a} \vee b \vee \bar{c})$.

5 Max Flow Approximation

Given that the previous FCFS flow problems are NP-hard, we could be interested by having some approximation algorithms. A “classic” flow is a relaxation of the First Come First Serve flow evaluation. One of the most famous optimization problem on the classic flow is the MAX FLOW problem which is polynomially solvable. MAX FLOW gives trivially an Upper Bound on many FCFS optimization problems (MAX FCFS FLOW TRIP PRICING...), however as we are going to show in this section, the approximation it gives can be really bad.

For instance because MAX FLOW doesn’t take into account reservation of parking spot at final destination it can be arbitrarily far from the optimal value. Figure 6 gives an example with two stations of capacity one and two vehicles where MAX FLOW would be able to serve q requests while any FCFS flow would not be able to serve any.

Now even if we consider unlimited station capacity, which means that we don’t have the problem of parking spot reservation anymore. As schemed in figure 7 MAX FLOW can again be arbitrarily far from the optimal value. Here with two stations, one vehicles and requests of maximum price 1, MAX FLOW would be able to serve q requests while any FCFS flow policy would not be able to serve more than one.

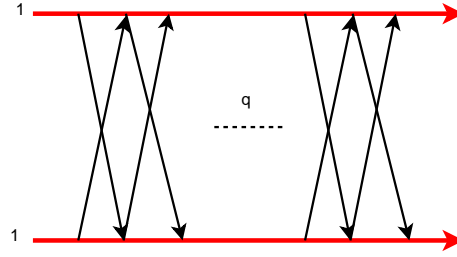


Figure 6: MAX FLOW doesn't consider parking spot reservation.

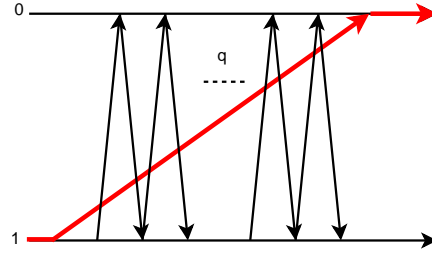


Figure 7: MAX FLOW is an unbounded approximation even with infinite station capacities.

Previous example used different transportation times for the same trip. Nevertheless with unique transportation time, and still infinite station capacity MAX FLOW is anyway at least a $\left(\frac{M \times (M-1)}{2}\right)$ -approximation (where M is the number of stations). Figure 8 schemes an instance with 3 stations, 1 vehicles and requests of maximum price 1 where MAX FLOW could serve $3 \times q$ requests while any FCFS flow policy could serve at most q . This example can be extended to M stations to find the $\left(\frac{M \times (M-1)}{2}\right)$ bound.

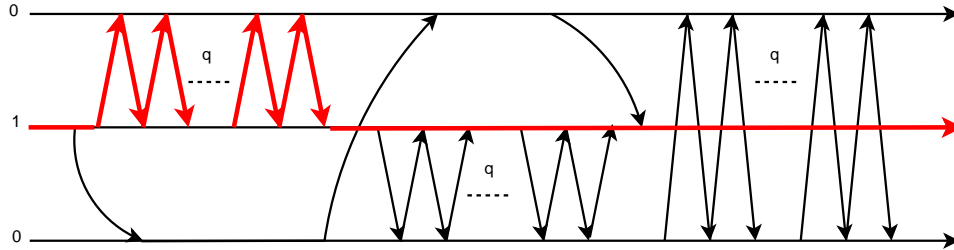


Figure 8: MAX FLOW gives at least a $\left(\frac{M \times (M-1)}{2}\right)$ -approximation even with infinite station capacity and unique transportation time.

In the end, we might observe in practice that MAX FLOW approximation is much better than a $\left(\frac{M \times (M-1)}{2}\right)$ -approximation so it could be still useful.

References

[1] <http://montreal.bixi.com>.

- [2] <http://www.autolib-paris.fr>.
- [3] <http://www.car2go.com>.
- [4] <http://www.velib.paris.fr>.
- [5] D. Chemla, F. Meunier, and R. Wolfler-Calvo. Balancing the stations of a self-service bike hiring system. 2011.
- [6] C. Contardo, C. Morency, and L-M. Rousseau. Balancing a dynamic public bike-sharing system. Technical Report 09, CIRRELT, 2012.
- [7] P. DeMaio. Bike-sharing: History, impacts, models of provision, and future. *Journal of Public Transportation*, 12(4):41–56, 2009.
- [8] C. Fricker and N. Gast. Incentives and regulations in bike-sharing systems with stations of finite capacity. 2012.
- [9] C. Fricker, N. Gast, and H. Mohamed. Mean field analysis for inhomogeneous bike sharing systems. 2012.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] C. Morency, M. Trpanier, and F. Godefroy. Insight into the montreal bikesharing system. 2010.
- [14] D.C. Shoup. *The High Cost of Free-Parking*. Planners Press, Chicago, 2005.
- [15] 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.
- [16] A. Wasserhole, V. Jost, and N. Brauner. Vehicle sharing system pricing regulation: A fluid approximation. 2012.