



HAL
open science

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-00727040v3

HAL Id: hal-00727040

<https://hal.science/hal-00727040v3>

Preprint submitted on 26 Apr 2013 (v3), 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 Optimization: Scenario-based approach

Ariel Waserhole^{1,2}

Vincent Jost²

Nadia Brauner¹

¹ G-SCOP, Grenoble University, France

² LIX CNRS, École Polytechnique Palaiseau, France

April 26, 2013

Abstract

One-way Vehicle Sharing Systems (VSS), such as Vélib' (2007), have a poor performance without regulation. We want to improve the efficiency of VSS using design strategies or pricing as incentive. Stochastic models are intractable for the size of systems we want to study. We therefore discuss a scenario-based approach, off-line deterministic optimization problem on a given stochastic realization (*scenario*). This deterministic model could be used to provide a heuristic and a bound for on-line stochastic optimization. This approach raises a new constraint the *First Come First Served constrained flow* (FCFS flow). We propose a design problem, MAX FCFS FLOW STATION CAPACITIES, and two pricing problems: MAX FCFS FLOW TRIP PRICING and MAX FCFS FLOW STATION PRICING based on a FCFS flow. We show that they are all APX-Hard. We study the upper bound given by the MAX FLOW problem and prove its poor worst case ratio to the MAX FCFS FLOW.

Keywords: Pricing; Vehicle Sharing Systems; Scenario-based approach; Queuing network; Complexity & approximation; Revenue Management; Graph vertex pricing.

1 Introduction

1.1 Context

Shoup (2005) reports that, based on a sample of 22 US studies, car drivers 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 (sharing the cost of ownership).

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 classic public transportation systems, short-term one-way VSS help to solve one of the most difficult public transit network design problem: the last kilometer issue (DeMaio, 2009). Round-trip VSS, where vehicles have to be returned at the station where they were taken, cannot address this important issue.

The first large-scale short-term one-way VSS was the bicycle VSS Vélib' (2007). 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

One way systems increase the user freedom at the expense of a higher management complexity. In round trip rental systems, while managing the yield, the only stock that is relevant is the number of available vehicles. In one-way systems, vehicles are not the only key resource anymore: parking stations may have limited number of spots and the available parking spots become an important control leverage.

Since first bicycle VSS, problems of bikes and parking spots availability have appeared very often. Reasons are various but we can highlight two important phenomenons: the gravitational effect which indicates that a station is constantly empty or full (as Montmartre hill in Vélib' (2007)), 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 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.

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) consider 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 objective is 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. The problem is to schedule truck routes to visit stations performing pickup and delivery. 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. (2012) and a dynamic one in Contardo et al. (2012).

1.3 A study on leverage for self regulated VSS

A new type of VSS has appeared lately: one-way Car VSS with Autolib' (2011) in Paris and Car2go (2008) in more than 10 cities (Vancouver, San Diego, Lyon, Ulm...). Due to the size of cars, operational balancing optimization through relocation with trucks seems inappropriate. Another way for optimizing the system has to be found.

This study is part of a work investigating different optimization leverage for self regulation in VSS. Using operation research we want to estimate the potential impact of:

- Optimizing the system design (station capacity, fleet size);
- Using pricing techniques to influence user choices in order to drive the system towards its most efficient dynamic;
- Establishing new protocols, for instance with parking spot reservations and/or users spatial and temporal flexibility.

Pricing policies can be *dynamic*, *i.e.* dependent on the system's state (vehicles distribution) or *static*, *i.e.* independent on system's state, set in advance and only function of the trip. In this paper, we focus on station capacities and static pricing problems. Note that station capacities can be seen as a dynamic pricing policy.

1.4 Scenario-based approach

In practice there is a lot of uncertainty in VSS dynamic. Dealing with human behavior, variability of user arrivals and transportation times has an important influence. In this context, stochastic optimization seems the most relevant approach to cope with randomness. Waserhole and Jost (2012) propose a stochastic one-way VSS pricing model based on closed queuing networks. For this model, a naive direct optimization with a Markov Decision Process computing the best dynamic (state dependent) policy is intractable: it can't even scale up for systems in the order of 7 stations. This problem is known as the curse of dimensionality; the number of states of the induced Markov Chain is exponential and hence exact solution techniques are not applicable. A fluid approximation that computes a static policy and an upper bound on potential optimization is discussed.

In this paper, we study a scenario-based approach for the VSS Pricing problem based on Wasserhole and Jost (2012) Closed Queuing Networks formulation. When dealing with stochastic problems, it is classic and natural to consider deterministic approximations. It amounts to optimizing a posteriori the system, considering that all trip requests (a *scenario*) are available at the beginning of the time horizon. Morency et al. (2011) show that, in Montreal BSS Bixi (2009), 68% of the trips were made by “members” and that their frequencies of use are quite stable along the week. For this context, considering deterministic requests might be a good approximation.

This approach offers two main advantages: On the one hand, the off-line deterministic optimization problem 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 (Bertsimas et al., 2011a), at least for models describing uncertainty by sets of scenarii.

Although this paper deals with VSS optimization, the theoretical problem addressed is the optimal control of closed queuing networks with general service time and arrival rate distributions. Therefore, our results can be applied to a wider class of queuing network problems to conduce performance analysis (Bertsimas et al., 2011b) or to estimate the relevancy of robust optimization.

The remaining of the paper is structured as follows: In Section 2, we describe a new type of constraint implied by the VSS scenario-based approach: the *First Come First Served constrained flow* (FCFS flow). In Section 3, we define a station capacity problem based on the FCFS constrained flow that is shown APX-hard. In Section 4, we define two pricing problems based also on this constraint that are both shown APX-hard: 1) The trip pricing problem that decides a price for taking each trip and 2) The station pricing problem that decides for each station the price to take and return a vehicle. Finally, in Section 5, we study a bound and an approximation algorithm for a FCFS flow pricing problems based on the MAX FLOW algorithm.

2 First Come First Served constrained flows

Vehicles moves can be modeled as a new type of constrained flow over a time and space network: the *First Come First Served constrained flow* (FCFS flow). Even if not explicitly specified nor named, this constraint is implicitly present in some continuous time models. For instance, it arises naturally in many applications such as in the fluid approximation of a Markov Decision Process (Maglaras, 2006; Wasserhole and Jost, 2012). However, to the best of our knowledge, the FCFS constrained flow is usually implicitly respected in continuous-time models and it has not been studied nor mentioned yet in discrete-time problems.

In the sequels, in order to remain in the lexical field of VSS, we speak about a flow of vehicles transiting among stations thanks to users. Nevertheless, in the more general context of queuing networks, it can be seen as a flow of clients moving along servers.

2.1 FCFS flow in time and space network

We consider a system of N vehicles transiting among a set S of stations with infinite capacities. The time horizon is $H = [0, T]$ and at time 0 the distribution of the vehicles among the stations is known. A trip request $r \in R$ asks for a vehicle from an origin station s_o^r at time t_o^r to a destination station s_d^r at time t_d^r . The vehicles move like an automatic flow, *i.e.* no decision can influence the moves. As time goes on, the vehicles transit between stations by accepting the first spatio-temporal trip requests they meet, hence applying the FCFS rule.

We can build a time and space network to 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. We assume that no two events occur exactly at the same instant. At time t , the trip request $r = (s_o^r, t_o^r = t, s_d^r, t_d^r) \in R$ is accepted if and only if there is a vehicle available at station s_o^r at this time. If trip request r is accepted, a vehicle is removed from station s_o^r and it will be available again at time t_d^r at station s_d^r . If the trip is rejected, nothing happens.

We call this process *First Come First Served constrained flow (FCFS flow)*. Figure 1 schemes an example of a FCFS flow with 3 stations, 12 requests and 2 vehicles, one available at station a and the other one available at station b at the beginning of the horizon. In this *scenario*, with 2 vehicles, only 5 trip requests among 12 are served.

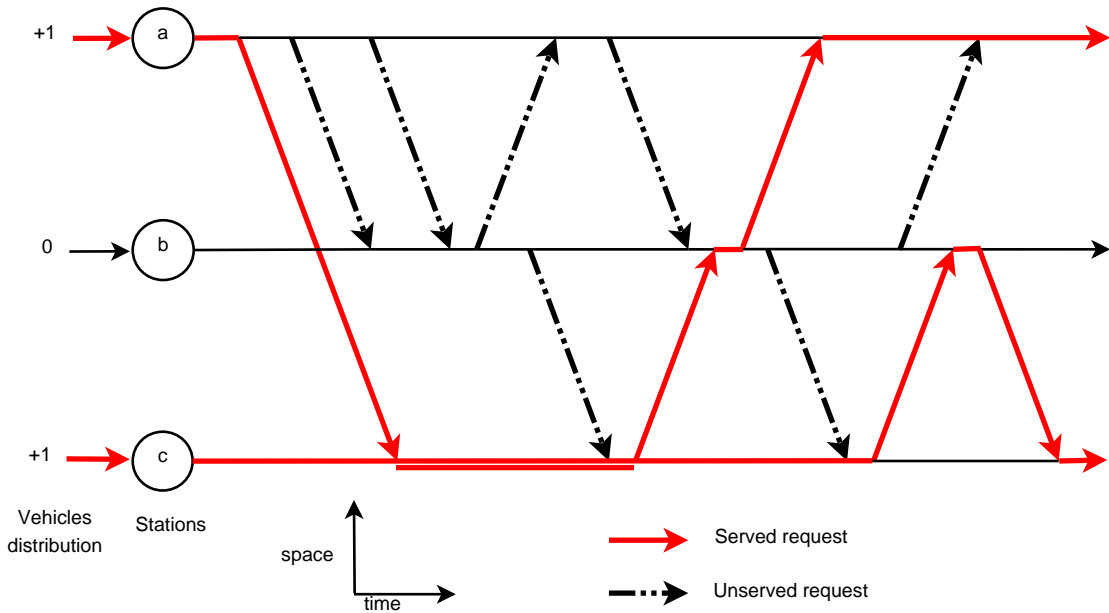


Figure 1: An example of a FCFS flow with 2 vehicles and 5 trip requests served.

2.2 Station capacity

If we consider now that station $s \in S$ has a capacity \mathcal{K}_s , blocking effect issues arise when a station is full. In theory, overbooking or client waiting time penalty might be interesting to

study. However in practice, in car VSS, users have the possibility to reserve a parking spot at destination to be sure to be able to retrieve the vehicle. Therefore, in order to avoid blocking effects, we assume that every trip is taken with a parking spot booked at destination.

To sum up, with station capacities, the trip request $r = (s_o^r, t_o^r = t, s_d^r, t_d^r) \in R$ 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 .

2.3 Priced FCFS flows

We now enhance the system with prices. A price p_{\max}^r is associated to request $r \in R$. This price is the maximum amount the user is willing to pay for taking the trip. The system proposes a fixed price $p_{a,b}$ for each trip $(a,b) \in S^2$. The set of requests that can be served is now reduced to $R_p = \{r \in R : p_{\max}^r \geq p_{s_o^r, s_d^r}\}$, namely the requests that can afford the price proposed by the system. If request r is accepted, it generates then a gain $p_{s_o^r, s_d^r}$. We call this process *priced FCFS flow*.

Figure 2 schemes an example of the run of such a process with 3 stations and 1 vehicle. The graph on the left represents the space network that indicates the prices proposed by the system. For this example, with 1 vehicle, 10 trip requests among the 12 can afford the asked price and finally 6 are served for a gain of 49.

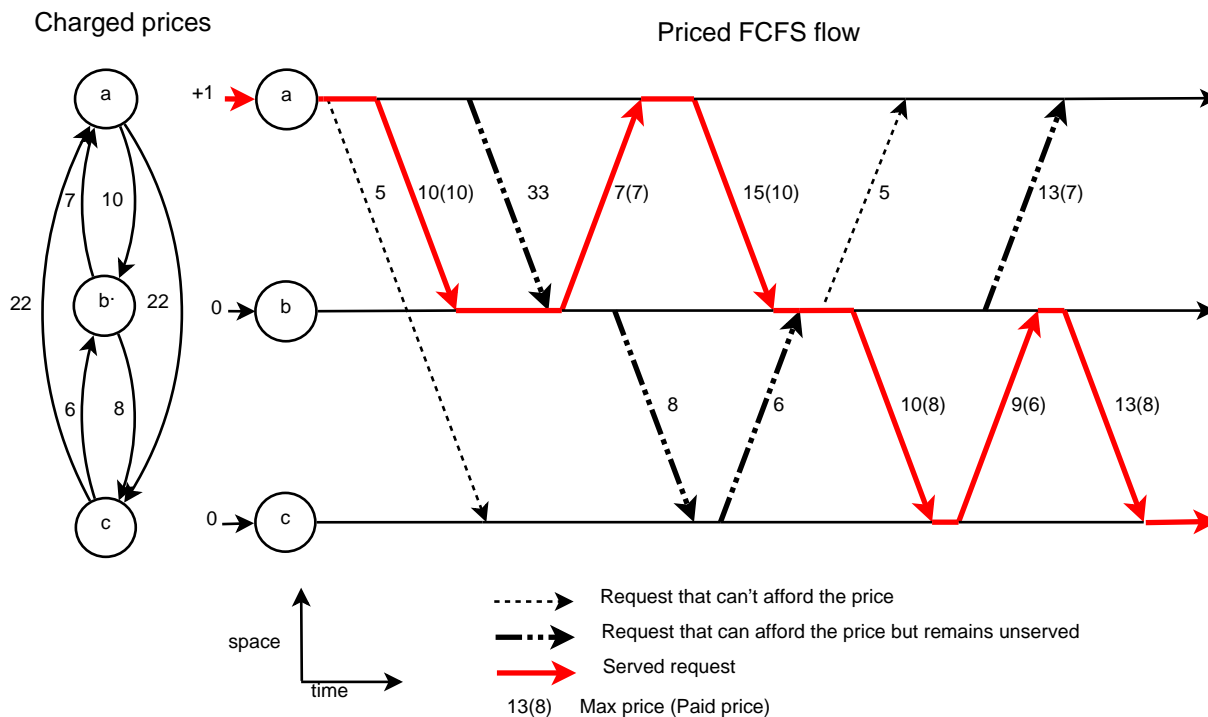


Figure 2: Priced FCFS flow with one vehicle and gain 49.

Finally, with station capacities, the trip request $r = (s_o^r, t_o^r = t, s_d^r, t_d^r, p_{\max}^r) \in R$ is accepted if

and only if there is a vehicle available at station s_o^r at time t , a parking spot available at station s_d^r also at time t and the user is willing to pay the proposed price, *i.e.* $p_{\max}^r \geq p_{s_o^r, s_d^r}$.

Remark 1 *The gain generated by a FCFS flow can be evaluated in linear time. Hence the decision versions of the optimization problems considered in the following are in NP.*

3 Station capacity problem

In this section we study the complexity of a tactical problem: setting a capacity for each station such that the number of trips sold in a FCFS flow for a set of trip requests with a unitary maximum price is maximized.

Intuitively, without any additional constraints, one would like to set all station capacities 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 tide phenomenons for instance. Station capacities are then used as a balancing tool. Figure 3 schemes an example of station capacity optimization. For this instance, the optimal capacity for station a is $\mathcal{K}_a = N/2$ while station b and c have a capacity $\geq N$. With this sizing, $N/2$ vehicles are taken by half of the trip requests from station b to station a at price 1 until station a is full. Then the remaining vehicles wait in station b before serving all trip requests going to station c at price 2. This policy generates the optimal final profit of $3N/2$ whereas setting all station capacities to N would lead to a profit of N .

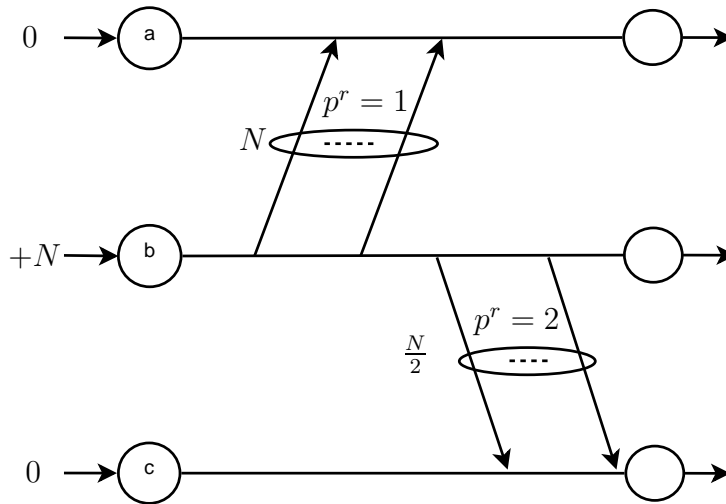


Figure 3: Example where proper station capacities increase the number of trips sold. Here setting $\mathcal{K}_a = N/2$ and $\mathcal{K}_b = \mathcal{K}_c \geq N$ gives the optimal revenue of $3N/2$.

We now formalize and derive some complexity results for this problem.

MAX FCFS FLOW STATION CAPACITIES

- **INSTANCE:** A set of stations S , a number N of vehicles with their distribution among the stations 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 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} .

Theorem 1 MAX FCFS FLOW STATION CAPACITIES *problem is NP-hard even with one vehicle.*

Proof: We reduce any instance (with n variables and m clauses) of the NP-complete problem 3-SAT (Garey and Johnson, 1979) to an instance of MAX FCFS FLOW STATION PRICING with one vehicle. Figure 4 schemes an example of such a reduction with two clauses. To each variable \dot{v} , we associate 3 stations \dot{v} , v and \bar{v} corresponding to the values unassigned, true and false. We define also two special stations res and tmp . The unique vehicle is located at station res at the beginning of the horizon.

All requests have a unitary maximum price 1 and they are built as follows: Each of the m clauses is taken iteratively. The first clause, let's say $a \vee \bar{b} \vee c$, contains variables \dot{a} , \dot{b} and \dot{c} . At time 1, there is a request from station res to the station representing the first variable \dot{a} . At time 2, the assignment of variable \dot{a} is modeled with two requests: from stations \dot{a} to a and then from \dot{a} to \bar{a} . At time 3, there is a request from the station representing the literal a contained in the clause to station res . Then, there is another request from station \bar{a} , representing the complement of the literal contained in the clause, to the station representing the next variable \dot{b} . At time 4, there are two successive requests, from station res to tmp and then from station tmp to res . At time 5, to treat the next variable \dot{b} , there is the same series of requests as in times 2, 3 and 4 but adapted to the current variable \dot{b} . At time 6, for the last variable of the clause \dot{c} , again, there is the same series of requests as in times 2, 3 and 4 adapted to this variable. However, this time, the last request returns to station res . This construction is then repeated for the next clauses.

For a given clause, in the time frame of its associated demands, the longest weighted path has a length and a gain equal to 9. There are 3 different longest weighted paths but all of them are starting and ending at station res . The maximum possible gain is then 9 and it is reached if and only if the assignment of variables satisfies the current clause. A MAX FCFS FLOW STATION CAPACITIES optimal solution for this instance can be transformed into a 3-SAT optimal solution and inversely with the relation: station a is closed and station \bar{a} is opened if, and only if, variable a is false. Finally 3-SAT has a solution if, and only if, the MAX FCFS FLOW STATION CAPACITIES built instance has a solution with gain $9m$. It proves that MAX FCFS FLOW STATION CAPACITIES is NP-hard for one vehicle. \square

Corollary 1 MAX FCFS FLOW STATION CAPACITIES *problem is APX-hard and not approximable within $39/40$ even with one vehicle.*

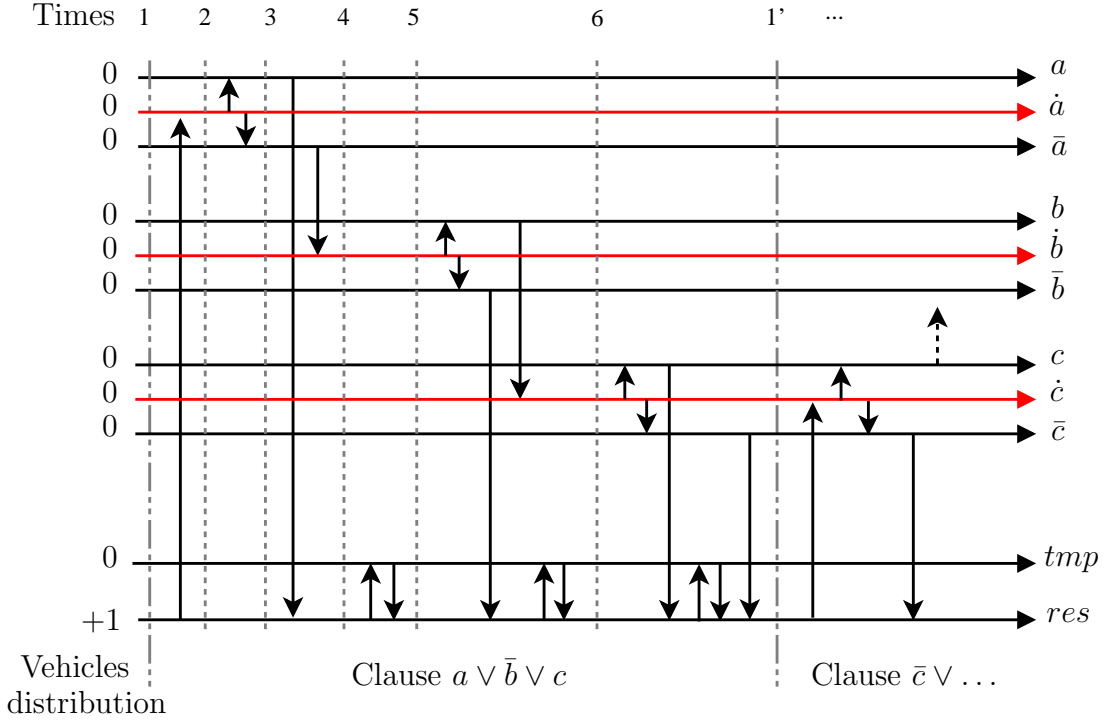


Figure 4: Reduction of 3-SAT to FCFS FLOW STATION CAPACITIES. Example with clauses $(a \vee \bar{b} \vee c) \wedge (\bar{c} \vee \dots)$.

Proof: MAX-3-SAT is the optimization problem associated to 3-SAT: given a 3-CNF formula, find an assignment that satisfies the largest number of clauses. We use the same construction as in the proof of Theorem 1 to reduce any MAX-3-SAT instance to a MAX FCFS FLOW STATION CAPACITIES instance with one vehicle. On the MAX FCFS FLOW STATION CAPACITIES instance, if a clause is not satisfied, the longest path possible is 7 and can always be obtained disregarding the variable assignment. Therefore, MAX-3-SAT has a solution with k clauses satisfied if and only if the MAX FCFS FLOW STATION CAPACITIES instance has a solution with gain $9k + 7(m - k) = 2k + 7m$.

Suppose that there exists an algorithm \mathcal{A} for the MAX FCFS FLOW STATION CAPACITIES problem giving a solution of value $F^{\mathcal{A}}$ with approximation ratio $\alpha \in [0, 1]$ from the optimal value F^* , i.e. $\frac{F^{\mathcal{A}}}{F^*} \leq \alpha$. For the instance built from MAX-3-SAT we have $F^{\mathcal{A}} = 2k^{\mathcal{A}} + 7m$ and $F^* = 2k^* + 7m$. Then:

$$\frac{2k^{\mathcal{A}} + 7m}{2k^* + 7m} \geq \alpha \quad \Leftrightarrow \quad 2k^{\mathcal{A}} \geq 2\alpha k^* + 7m(\alpha - 1). \quad (1)$$

A 3-SAT instance always admits a variable assignment satisfying at least $7/8$ of the clauses (Karloff and Zwick, 1997), i.e. $k^* \geq \frac{7}{8}m$. Since $1 - \alpha \geq 0$ we have $m(\alpha - 1) \geq \frac{8}{7}k^*(\alpha - 1)$ and then:

$$(1) \Rightarrow \frac{k^{\mathcal{A}}}{k^*} \geq 5\alpha - 4. \quad (2)$$

MAX-3-SAT is not approximable within $7/8$ unless $P=NP$ (Karloff and Zwick, 1997) therefore $\frac{k^A}{k^*} \leq \frac{7}{8}$ and we finally have:

$$(2) \Rightarrow 5\alpha - 4 \leq \frac{7}{8} \Leftrightarrow \alpha \leq \frac{39}{40}.$$

Hence MAX FCFS FLOW STATION CAPACITIES is not approximable within $39/40$ unless $P=NP$. \square

4 Pricing problems

In section 3 we discussed the complexity of a tactical problem, the design of the station capacities. We now study the complexity of an operational problem: the system management optimization through price leverage. We are searching for pricing policies maximizing the gain of the induced priced FCFS flow.

This investigation leads to the definition of two optimization problems which are both shown APX-Hard: the trip pricing problem which sets a price for each origin-destination pair independently and the station pricing problem which sets, for each station, a price for taking and a price for returning a vehicle. Note that the complexity results can be extended to *time dependent prices* (as long as prices remain constant on some time intervals). Time dependent prices allow to have different prices in the morning, middle of day and evening in order to control the tide phenomenon for instance.

4.1 FCFS Flow Trip Pricing problem

We 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.

MAX FCFS FLOW TRIP PRICING

- **INSTANCE:** A set of stations S with capacities \mathcal{K}_s for $s \in S$, a number N of vehicles with their 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 .

To study MAX FCFS FLOW TRIP PRICING complexity, we extend the approach used for MAX FCFS FLOW STATION CAPACITIES in the previous section.

Theorem 2 MAX FCFS FLOW TRIP PRICING *problem is APX-hard and not approximable within $39/40$, even with one vehicle and unitary maximum prices.*

Proof: We reduce a MAX-3-SAT instance to a MAX FCFS FLOW TRIP PRICING instance with one vehicle with the same reduction as in the proof of Theorem 1. Moreover, we consider that all requests have a unitary maximum price: *i.e.* $p_{\max}^r = 1, \forall r \in R$. There is a bijection between an optimal MAX-3-SAT solution and an optimal MAX FCFS FLOW TRIP PRICING solution for this instance with the following relation: trips to station a are closed, *i.e.* $p_{b,a} = \infty, \forall b \in S$, and trips to station \bar{a} are open, $p_{b,\bar{a}} = 1, \forall b \in S$, if and only if variable a is false. Finally, the proof of Corollary 1 can be applied again to show that MAX FCFS FLOW TRIP PRICING is not approximable within 39/40 unless P=NP. \square

Remark 2 *If a FCFS flow problem is hard even for one vehicle, then it is also hard for infinite station capacities. Therefore MAX FCFS FLOW TRIP PRICING is APX-hard even with infinite capacities.*

4.2 FCFS Flow Station Pricing problem

We now consider 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, using for example a price heated maps as in Papanikolaou (2011): stations are colored depending on their prices, for instance from yellow for cheap to red for expensive.

We study the complexity of MAX FCFS FLOW STATION PRICING. Without loss of generality, we consider that prices are independent from the distance/time the vehicle is used. We show that this problem is already *hard* in the single choice context, *i.e.* users only have one possibility for origin/destination.

MAX FCFS FLOW STATION PRICING

- **INSTANCE:** A set of stations S with capacities K_s for $s \in S$, a number N of vehicles with their 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 3 *MAX FCFS FLOW STATION PRICING is APX-HARD and not approximable within 39/40 even with one vehicle or infinite station capacities.*

Proof: We reduce a MAX FCFS FLOW TRIP PRICING instance (TRIP-INST) to a MAX FCFS FLOW STATION PRICING instance (STATION-INST).

STATION-INST is composed with the same set of stations as TRIP-INST plus 2 new stations, ab^1 and ab^2 , for each possible trip (a, b) . For each trip request $r = (s_o^r = a, t_o^r, s_d^r = b, t_d^r, p_{\max}^r)$ of TRIP-INST, STATION-INST has 3 trip requests: $(a, t_o^r, ab^1, t_o^r + \epsilon, 0)$, $(ab^1, t_o^r + 2\epsilon, ab^2, t_o^r + 3\epsilon, p_{\max}^r)$ and $(ab^2, t_o^r + 4\epsilon, b, t_d^r, 0)$, with ϵ such that $0 < 4\epsilon < t_d^r - t_o^r$.

Note that STATION-INST solutions with $p_t(a) = p_r(ab^1) = p_t(ab^2) = p_r(b) = 0, \forall a, b \in S$ are dominant. Moreover, there is a transformation respecting the objective value between an optimal TRIP-INST and an optimal STATION-INST with the relation $p_{a,b} = p_t(ab^1) + p_r(ab^2)$ for each possible trip (a, b) . TRIP-INST has a solution of gain at least g if and only if STATION-INST has a solution of gain at least g . Theorem 2 proves that MAX FCFS FLOW TRIP PRICING is APX-hard and not approximable within $39/40$ even with one vehicle, therefore MAX FCFS FLOW STATION PRICING is also APX-hard with the same ratio. As in Remark 2, it is also APX-hard for infinite station capacities. \square

4.3 FCFS flow relaxation: GRAPH VERTEX PRICING

In Theorem 3 we show that MAX FCFS FLOW TRIP PRICING can be reduced to MAX FCFS FLOW STATION PRICING. The opposite reduction doesn't seem trivial. In fact, there is another difficulty in MAX FCFS FLOW STATION PRICING not linked with the flow constraint: the quadratic price assignment. We shall therefore consider a simplified version of the problem where we remove the flow constraint, namely the MAX ORIENTED GRAPH VERTEX PRICING (O-GVP) problem and its unoriented version MAX GRAPH VERTEX PRICING (GVP). They are simplifications of MAX FCFS FLOW STATION PRICING. We prove that they are already both APX-hard.

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

MAX ORIENTED GRAPH VERTEX PRICING (O-GVP)

- **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:

$$\sum_{\substack{(a,b) \in A \\ / p_t(a) + p_r(b) \leq c_{a,b}}} p_t(a) + p_r(b).$$

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

MAX GRAPH VERTEX PRICING (GVP)

- **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:

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

GVP has already been studied in the literature. It is one of the fundamental special cases of the Single-Minded item Pricing (SMP) problem (Guruswami et al., 2005). Khandekar et al. (2009) prove that GVP is APX-hard on bipartite graphs. The best known approximation algorithm, by Balcan and Blum (2006), gives a 4-approximation. We now present a polynomial reduction from GVP to O-GVP to show that the latter is also APX-hard.

Theorem 4 MAX ORIENTED GRAPH VERTEX PRICING *is APX-hard even on bipartite graphs.*

Proof: We reduce a GVP instance to a O-GVP instance. GVP is APX-hard even on bipartite graphs (Khandekar et al., 2009). 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. On this oriented graph, O-GVP solves GVP. Hence, O-GVP is APX-hard even on bipartite graph. \square

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

Corollary 2 MAX FCFS FLOW STATION PRICING *is APX-hard even with an unlimited number of vehicles, infinite station capacities or 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 O-GVP taking each request as an arc. MAX ORIENTED VERTEX PRICING is shown NP-hard on bipartite graphs, therefore MAX FCFS FLOW STATION PRICING is APX-hard even with requests defining a bipartite graph. \square

5 Relationships with the MAX FLOW problem

Given that the FCFS flow problems presented in the previous sections are APX-hard, bounds or approximation algorithms might be of interest. A “classic” flow is a relaxation of the first come first served flow evaluation. One of the most famous optimization problem on classic flows is MAX FLOW which is polynomially solvable. MAX FLOW gives an Upper Bound (UB) on many

FCFS optimization problems such as MAX FCFS FLOW STATION CAPACITIES or MAX FCFS FLOW TRIP/STATION PRICING.

In practice, Waserhole and Jost (2012) observe that the ratio between the MAX FLOW and the FCFS flow optimization problem is roughly in a factor 2. In Section 5.1, we show that the theoretical guaranty (worst case) of this UB is extremely poor. In Section 5.2, we refine on the MAX FLOW UB through an approximation algorithm for the FCFS FLOW 0/1 TRIP PRICING, *i.e.* the FCFS FLOW TRIP PRICING with unitary maximum prices.

5.1 MAX FLOW upper bounds for FCFS flow problems

MAX FLOW Classic flows don't take into account reservation of parking spot at the destination station, therefore MAX FLOW gives an UB that can be arbitrarily far from any FCFS flow. Figure 5 schemes an example with 2 stations of unitary capacity and 2 vehicles with q crossed demands. In this example, MAX FLOW is able to serve all q requests while any FCFS flow with reservation can't serve any.

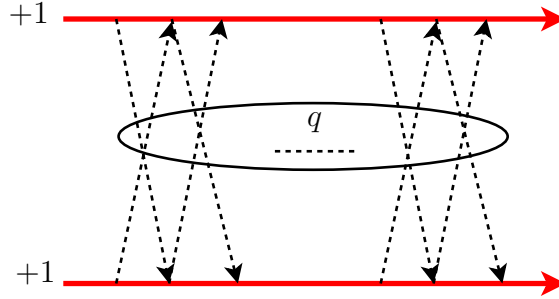


Figure 5: MAX FLOW UB can be arbitrarily far from any FCFS flow since it doesn't consider parking spot reservation.

MAX FLOW WITH RESERVATION Assuming that no two requests arrive at the same time, we can add constraints to the MAX FLOW classic linear program to respect parking spot reservations. As schemed in Figure 6, it amounts to considering requests with null transportation time, respecting station capacities, and then a time where the vehicle is unavailable at the station. The case represented Figure 5 is then avoided. We call this problem MAX FLOW WITH RESERVATION (MAX FLOW WR). MAX FLOW WR remains polynomial. However, solving it with a classic linear programming solver is much slower than MAX FLOW because classic flow algorithms do not apply anymore (Waserhole and Jost, 2012).

MAX FLOW WR can again be arbitrarily far from any FCFS flow. Figure 7 schemes it on an example with 2 stations, Lower (L) and Upper (U), 1 vehicle available at L at the beginning of the horizon and trip requests with unitary maximum prices. The first request goes from L to U and takes the entire horizon to reach the station U. Then there are q successive trip requests from L to U and from U to L. In this instance, MAX FLOW WR is able to serve q requests,

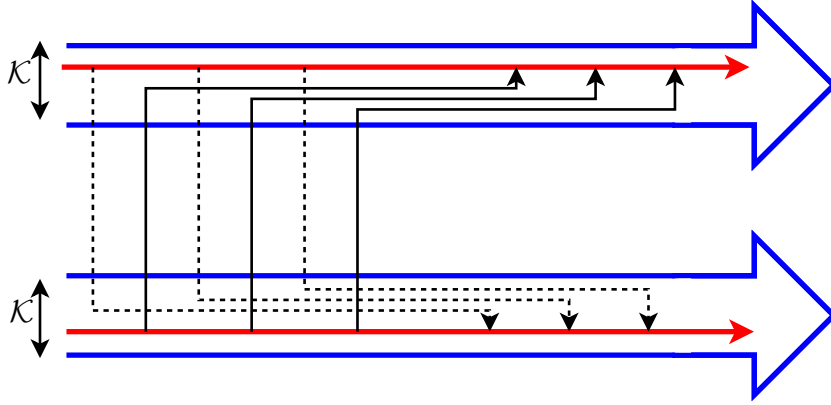


Figure 6: A MAX FLOW WITH RESERVATION.

rejecting only the first long one, while any FCFS flow can't serve more than one request, the first one.

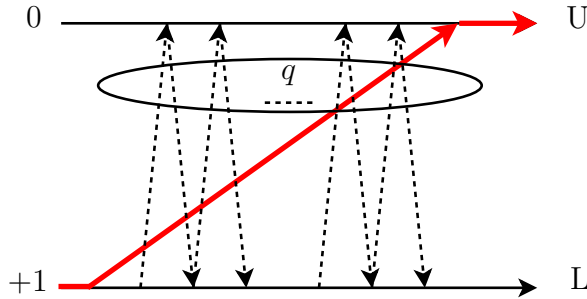


Figure 7: MAX FLOW WITH RESERVATION UB gap is unbounded.

MAX FLOW WR for non-crossing requests The previous example used *crossing requests* for the same trip: *i.e.* one request asks for a trip within the transportation time-frame of another one for the same trip. For instance, unitary transportation times imply *non-crossing requests*. With non-crossing requests, MAX FLOW WR can still be $2^M - M - 1$ times better than any feasible FCFS flow, where M is the number of stations.

For one vehicle and a given number of stations M , an instance reaching the $2^M - M - 1$ bound can be constructed as follows: The instance is based on a succession of repeated cyclic requests. A cyclic request is an ordered series of trip requests evolving along a cycle in the physical graph of stations. There are $2^M - M - 1$ different station disjoint cycles and hence $2^M - M - 1$ different cyclic requests (we do not take the empty cycle nor cycles with only one station). Each cyclic request is repeated to have a total of q trip requests. The stations present in a cyclic request are called the support. Before each repetition of the same cyclic request, the entrance is forced into one specific station of the support, say s_1 , thanks to a gadget that creates a request from every station to s_1 . Then starts the first cyclic request that is special. It begins with s_1 and before each trip request of the cyclic request, there are a series of requests from its current origin

station going out to every station not present in the support. The cyclic request is then repeated in order to contain in the end q trip requests. With one vehicle, on this instance, MAX FLOW can serve $(2^M - M - 1)q$ demands while any FCFS flow policy can serve at most $q + O(2^M)$. Asymptotically, when q tends to infinity, the gap between MAX FLOW WR and any FCFS flow tends to $2^M - M - 1$. For $M = 5$ stations, Figure 8 schemes how to create the requests for one repeated cyclic request which support is the set of 3 stations a , b and c .

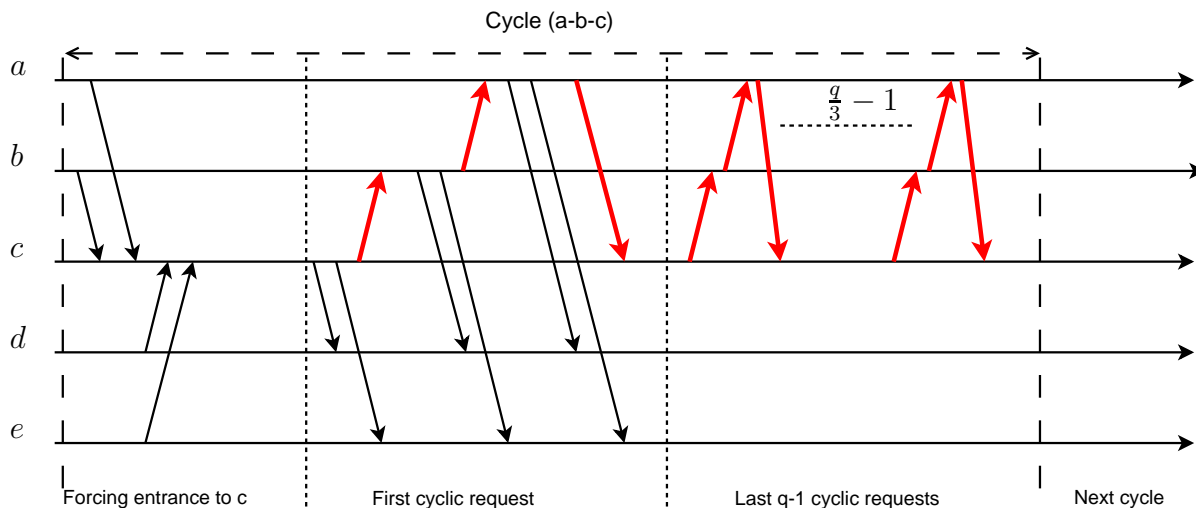


Figure 8: For non-crossing requests, the gap between MAX FLOW WITH RESERVATION and any FCFS flow is greater than $2^M - M - 1$.

5.2 An approximation algorithm for FCFS FLOW 0/1 TRIP PRICING

For non crossing requests, we present now an approximation algorithm for FCFS FLOW 0/1 TRIP PRICING (FCFS FLOW TRIP PRICING with unitary maximum prices). To do so, first, we give an approximation algorithm for FCFS PATH 0/1 TRIP PRICING which is the FCFS FLOW 0/1 TRIP PRICING problem with one vehicle. This approximation algorithm is based on the MAX FLOW optimal solution. It returns a *cyclic policy* that we define as a policy that can serve only trip requests belonging to one oriented cycle in the spatial network.

Theorem 5 *Algorithm 1 provides an $(M + 2)!$ approximation algorithm for the FCFS PATH 0/1 TRIP PRICING problem with non-crossing requests.*

Proof: Algorithm 1 gives, for each detected cycle c , its occurrence $n(c)$ and its length $|c|$ in the Max Flow optimum solution for one vehicle. Figure 9 schemes an example of execution with 2 detected cycles each one appearing once. Each cycle has a length greater or equal to 2 and between two consecutive cycles we can't iterate through more than $M - 2$ stations (*lost requests*). It means that every M stations we detect at least a cycle of size 2. Hence, keeping

Algorithm 1 FCFS PATH 0/1 TRIP PRICING Greedy Approximation Algorithm

```

1:  $F^* \leftarrow$  MAX FLOW solution for 1 vehicle in the time & space network;
2: for all Station  $s$  in path  $F^*$  do ▷ Iterate on path  $F^*$ 
3:   if  $s$  is marked then ▷ A cycle  $c$  (starting and ending at  $s$ ) is detected
4:      $n(c) \leftarrow n(c) + 1$ ;
5:     Unmark all stations;
6:   end if
7:   Mark station  $s$ ;
8: end for
9: return the cyclic policy defined by the cycle  $c$  with maximum value  $n(c)|c|$ .

```

only the detected cycles might lose a factor at most $2/M$:

$$\sum_c n(c)|c| \geq \frac{2}{M} F^*.$$

Moreover, there are less than $M \times M!$ different cycles, therefore the cycle c' with the maximum $n(c)|c|$ has for value at least:

$$n(c')|c'| \geq \frac{2}{M \times M \times M!} F^* \geq \frac{1}{(M+2)!} F^*.$$

c' defines a cyclic policy C' that provides at least the same gain ($C' \geq n(c')|c'|$) with a FCFS flow dynamic and all requests (assumed non-crossing). Finally, Algorithm 1 is polynomial, therefore for non-crossing requests we have a $(M+2)!$ approximation on the optimal FCFS PATH 0/1 TRIP PRICING policy S^* :

$$(M+2)! C' \geq F^* \geq S^*.$$

□

We now extend the preceding FCFS path results to the FCFS flow problem.

Corollary 3 *For non-crossing requests we have the following results:*

- *Algorithm 1 provides an $N(M+2)!$ approximation algorithm for the FCFS FLOW 0/1 TRIP PRICING problem.*
- *The approximability ratio of the FCFS FLOW 0/1 TRIP PRICING is within $[39/40, N(M+2)!]$.*
- *The theoretical gap (worst case) between MAX FLOW WR and any FCFS flow is within $[2^M - M - 1, N(M+2)!]$.*

Proof: We assume non-crossing requests. Theorem 2 states that FCFS FLOW TRIP PRICING is not approximable within $39/40$ even with unitary maximum prices, that is FCFS FLOW 0/1 TRIP PRICING.

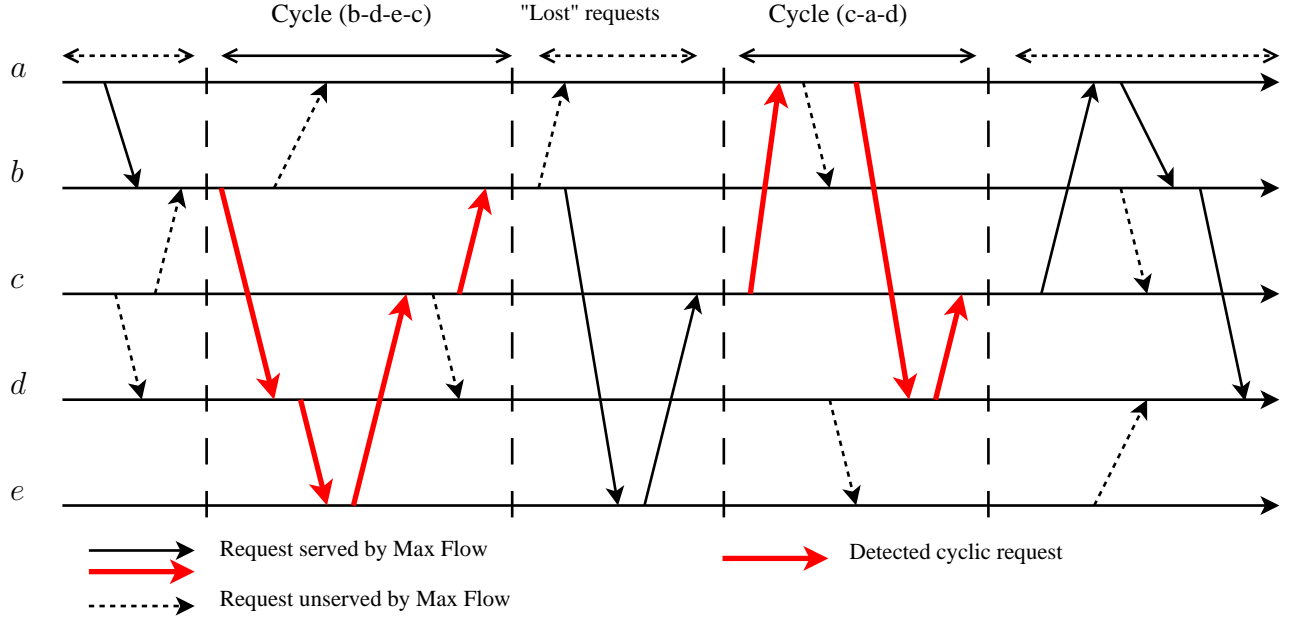


Figure 9: Example of execution of the Greedy Algorithm 1 where two cycles are detected with occurrence 1.

Theorem 5 can be extended to any number of vehicles. Let F_1^* be the value of MAX FLOW for 1 vehicle and F_N^* for N vehicles. We have $NF_1^* \geq F_N^* \geq S^*$ and hence $N(M+2)!C' \geq S^*$. Therefore, Algorithm 1 provides an $N(M+2)!$ approximation algorithm for the FCFS FLOW 0/1 TRIP PRICING problem and, unless P equals NP, FCFS FLOW 0/1 TRIP PRICING approximability ratio is within $[39/40, N(M+2)!]$.

Let F_N^{R*} be the value of MAX FLOW WR for N vehicles. In the proof of Theorem 5, we saw that $C' \geq \frac{1}{(M+2)!}F_1^*$. Since $S^* \geq C'$ and $NF_1^* \geq F_N^* \geq F_N^{R*}$ we have: $N(M+2)!S^* \geq F_N^*$. Moreover, we have seen in the previous section that there exists instances such that $F_1^{R*} \geq (2^M - M - 1)S^*$. Therefore, the theoretical gap between a MAX FLOW WR and any FCFS flow is within $[2^M - M - 1, N(M+2)!]$. \square

6 Conclusion

In this paper, we investigate a scenario-based approach for a stochastic VSS model. Its principle is to work a posteriori on a realization of the stochastic process: *a scenario*. Optimizing on a scenario provides heuristics and bounds for the stochastic problem. In this context, such approximation raises deterministic problems with a new constraint: the *First Come First Served constrained flow* (FCFS flow). We present three such problems, a system design problem, optimizing station capacity: FCFS FLOW STATION CAPACITIES; two static pricing problems, considering a price to take each trip: FCFS FLOW TRIP PRICING, or considering that the price to take a trip is the addition of the price to take a vehicle in a station and to return it in another:

FCFS FLOW STATION PRICING.

We show that all three problems are APX-hard, *i.e.* inapproximable in polynomial time within a constant ratio. Therefore, we investigate a bound and an approximation algorithm using the MAX FLOW algorithm (hence relaxing the FCFS flow constraint). The theoretical guaranty (worst case) for the bound provided by the MAX FLOW algorithm on a scenario is exponential in the number of station. Nevertheless, it seems competitive in practice (Waserhole and Jost, 2012), and it can be used to build an $N(M + 2)!$ approximation algorithm for the FCFS FLOW 0/1 TRIP PRICING problem; with N the number of vehicles and M the number of stations.

We conjecture that the inapproximability ratios of FCFS FLOW TRIP/STATION PRICING and FCFS FLOW STATION CAPACITIES are greater than a factor linked to the number of stations. One can hence be satisfied to have an approximation algorithm that does not depend on the number of trip requests $|R|$. However, in current VSS, the number of trips sold in one day is in the order of M (or N). Therefore, an approximation algorithm in $|R|$ might be more useful.

Finally, giving good and usable heuristic solutions using scenario-based optimization, studying metaheuristic approaches might be interesting. However, it is not sure that they can explore a large space and provide good solutions within a reasonable time. Indeed, the evaluation cost of a *movement* on a static policy seems important, at first sight basically in the order of computing again the whole FCFS flow.

References

- Autolib'. <http://www.autolib-paris.fr>, 2011.
- M-F. Balcan and A. Blum. Approximation algorithms and online mechanisms for item pricing. In *Proceedings of the 7th ACM Conference on Electronic Commerce*, pages 29–35. ACM, 2006.
- D. Bertsimas, D. B. Brown, and C. Caramanis. Theory and applications of robust optimization. *SIAM Rev.*, 53(3):464–501, August 2011a. ISSN 0036-1445. doi: 10.1137/080734510.
- D. Bertsimas, D. Gamarnik, and A. A. Rikun. Performance analysis of queueing networks via robust optimization. *Operations Research*, 59(2):455–466, 2011b.
- Bixi. <http://montreal.bixi.com>, 2009.
- Car2go. <http://www.car2go.com>, 2008.
- D. Chemla, F. Meunier, and R. Wolfler Calvo. Bike sharing systems: Solving the static rebalancing problem. *Discrete Optimization*, 2012.
- 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.
- V. Guruswami, J.D. Hartline, A.R. Karlin, D. Kempe, C. Kenyon, and F. McSherry. On profit-maximizing envy-free pricing. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1164–1173. Society for Industrial and Applied Mathematics, 2005.
- H. Karloff and U. Zwick. A 7/8-approximation algorithm for max 3sat? In *Foundations of Computer Science, 1997. Proceedings., 38th Annual Symposium on*, pages 406 –415, oct 1997.
- R. Khandekar, T. Kimbrel, K. Makarychev, and M. Sviridenko. On hardness of pricing items for single-minded bidders. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 202–216, 2009.
- 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.
- C. Maglaras. Revenue management for a multiclass single-server queue via a fluid model analysis. *Operations Research*, 2006.
- C. Morency, M. Trépanier, and F. Godefroy. Insight into the Montreal bikesharing system. In *TRB-Transportation Research Board Annual Meeting, Washington, USA, Paper #11-1238, 17 pages*, January 2011.
- D. Papanikolaou. The market economy of trips. Master’s thesis, Massachusetts Institute of Technology, 2011.
- 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.
- Vélib’. <http://www.velib.paris.fr>, 2007.
- A. Wasserhole and V. Jost. Vehicle sharing system pricing regulation: A fluid approximation. 2012. URL <http://hal.archives-ouvertes.fr/hal-00727041>.