Abstract—Managing a one-way vehicle sharing system means periodically moving free access vehicles from excess to deficit stations in order to avoid local shortages. We perform a lower bound analysis for the static version of the resulting operational decision problem, and derive from this analysis two heuristic algorithms whose main feature is to be vehicle oriented, which means that they focus on the way vehicles are exchanged between excess and deficit stations.

I. INTRODUCTION

VEHICLE Sharing systems [13, 16] are emerging mobility systems which aim at compromising between purely individual mobility and rather rigid public transportation. Such a system is composed of a set of stations, at which free access vehicles are parked. Those vehicles can be bicycles or electric cars. There exists a special station called Depot, in which a set of carriers (trucks, self-platoon convoys, ...) are stored, which periodically exchange vehicles between the stations and eventually provide the system with additional vehicles. A trend is to make the system be a one-way system, which means that users are not imposed to give vehicles back at the station where they have been picking up. This feature makes the system more attractive. But a drawback is that it raises the eventuality of unbalanced situations, in the sense that some stations may become overfilled other underfilled, provoking local shortages or making users unable to give their vehicle back. In order to avoid such a situation, managers have to periodically perform a relocation process: carriers pick up vehicles from excess stations and transfer them to deficit stations. Performing this process while meeting both economic and quality of service purposes means addressing a Vehicle Sharing Relocation problem (VSR). Though practically this VSR problem has to be handled on line [13, 14], most related academic studies have been involving static (see [5, 6, 13, 15, 19]), or eventually dynamic formulations [11, 17].

Those formulations, which differ in a significant way from an author to another, have been mostly handled through hierarchical decomposition into a carrier routing master model and a vehicle load/unload slave model, and through local search or genetic algorithms (see [6, 8, 10, 18]). Their common feature is that they are carrier oriented, in the sense that they focus on the construction of the recollection tours which are run by the carriers, and consider the routing of the vehicles inside the carriers as a kind of slave object. Such an approach may be discussed because it cannot rely on a backward link between the master carrier tour collection and the vehicle sub-problem, which would provide sensitivity information and help in driving the search for the master object. It comes that the search for the master carrier tour collection is very often performed in a somewhat blind way.

We adopt here the opposite point of view and consider that an efficient way to perform a relocation process is to route the vehicles from excess stations to deficit ones in a way which make them share, as often as possible, related carriers. So the purpose of this work is to propose and test alternative approaches to carrier oriented ones, which we shall call vehicle oriented: the vehicle routing strategy becomes the master object, which determines in turn the carrier routes.

The paper is organized as follows. First we introduce a formal VSR model, generic in the sense that it mixes different criteria: economic cost of the relocation process (number of carriers and carrier riding time), and quality of service (unavailability of the vehicles during the process). Next we perform a lower bound analysis of this VSR model. The way we obtain lower bounds leads us to the design of 2 heuristic VSR algorithm: the first one considers the way vehicles are distributed from excess stations to deficit ones as the master object and relies on a Min Cost Assignment/Pick up and Delivery decomposition; the second one considers the aggregated routing of the vehicles along the station network as a main object, and relies on a lift procedure which turns an aggregated routing of vehicles and carriers into a feasible VSR solution. We end with numerical experiments.
II. THE VSR MODEL

A. Instances, Feasible Solutions and Models.

VSR (Vehicle Sharing Relocation Problem) Instances: We consider here a set $X$ of stations, one of them being a specific station Depot. Any station $x$ is provided with a coefficient $v(x)$, which tells us that $v(x)$ vehicles are in excess at station $x$: if $v(x)$ is strictly negative, then we need to bring $-v(x)$ vehicles to station $x$ (a station $x$ is then said to be a deficit station); if $v(x)$ is strictly positive, then $x$ is an excess station and we need to remove $v(x)$ from $x$; if $v(x) = 0$ then $x$ is said to be neutral. We suppose that $\sum_{x \in X} v(x) = 0$, which means that the Depot station may be used to bring additional vehicles to the system, or, conversely, to remove some of them. $DIST$ denotes the $X.X$ distance matrix: $DIST_{x,y}$ is the distance (time required for a carrier to go from $x$ to $y$) between station $x$ to station $y$. Matrix $DIST$ is not required to be symmetric, but should satisfy the Triangle inequality. $T$.Max is the maximal makespan of the relocation process, which means that the total duration of this process should not exceed $T$.Max. By the same way, $COST$ denotes the carrier cost matrix: $COST_{x,y}$ is the cost which is induced for a carrier when it moves from $x$ to $y$. All carriers are identical with capacity $CAP$ and initially located at the Depot station. This defines a VSR instance $(X, v, CAP, T$.Max, $DIST, COST)$. 

VSR Feasible Tours: A VSR tour $\Gamma$ is a finite sequence $\Gamma_{Route} = \{x_0 = Depot, x_1, \ldots, x_{n(\Gamma)} = Depot\}$ of stations, which is called a route, given together with a loading strategy, that means with 2 sequences $\Gamma_{Load} = \{L_0, L_1, \ldots, L_{n(\Gamma)}\}$ and $\Gamma_{Route} = \{T_0 = 0, T_1, \ldots, T_{n(\Gamma)}\}$ of coefficients whose meaning is: a carrier which follows the route $\Gamma$ loads, at time $T_i$, $L_i$ vehicles at station $x_i$ (unloads in case $L_i < 0$). The $COST$-length $L$.COST($\Gamma$) of such a tour is the sum $\sum_j (COST_{x_j, x_{j+1}})$ and its $DIST$-length $L$.DIST($\Gamma$) is the sum $\sum_j (DIST_{x_j, x_{j+1}})$. This VSR tour $\Gamma$ is VSR feasible if:

- For any $i = 0, \ldots, n(\Gamma)-1$, $T$.Max $\geq T_{i+1} - T_i + DIST_{x_i, x_{i+1}}$.
- For any $i = 0, \ldots, n(\Gamma)-1$, $0 \leq L_i = \sum_{j=0}^{i-1} L_j \leq CAP$.
- For any $j$ such that $v(x_j) \geq 0$, then $v(x_j) \geq L_j \geq 0$.
- For any $j$ such that $v(x_j) \leq 0$, then $v(x_j) \leq L_j \leq 0$.

Explanation: (E1): A carrier needs at least $DIST_{x_i, x_{i+1}}$ time units in order to go from $x_i$ to $x_{i+1}$; (E2): $L_j$ denotes its current load when it leaves $x_j$ and this loads cannot exceed the capacity $CAP$; (E3): Any carrier is empty when its starts and finishes a tour; (E4, E5): loading (unloading) operations are respectively restricted to excess (deficit) stations, which also means that we impose a given vehicle to be moved from an origin station to a destination station by exactly one carrier (Non Preemption hypothesis).

Given scaling coefficients $\alpha, \beta, \delta$, together with a VSR instance $(X, v, CAP, T$.Max, $DIST)$, we set:

VSR Model: $\{\text{Compute a VSR feasible tour collection } \Gamma^* = \{\Gamma(k), k = 1..K\} \text{ such that:}\$

- For any station $x$, $\sum_{e \in \Gamma(k)} l(e) \leq \alpha.K + \beta.COST(\Gamma(k)) + \delta(\sum_{x \in X} v(x) - \sum_{e \in \Gamma(k)} l(e)(x))$.  

Explanation: (E6): For any excess station $x$, $v(x)$ vehicles have to be picked up in $x$, and for any deficit station $x$, $-v(x)$ vehicles have to be delivered to $x$.

Minimize: $\text{Cost}(\Gamma^*) = \text{a weighted sum of the active carrier number } \alpha.K$, the carrier riding cost $\sum_{x \in X} v(x)$ and the vehicle riding time (time vehicles spend into the carriers) $\sum_{x \in X} (L$.COST($\Gamma(k)$)).

Remark 1 about MIP VSR models and Complexity: Modeling VSR through a MIP (Mixed Integer Linear Program) is possible, but difficult and inefficient. As for complexity, VSR is NP-Hard, even if we consider one carrier ($\alpha$ very large) with capacity $1$, if every quantity $v(x)$ is equal to $1$ or $-1$, and if $\delta = 0$. In such a case, solving the problem becomes equivalent to solving the Travelling Salesman problem on a bipartite graph (the excess stations on one side and the deficit ones on the other side), which is known to be NP-Hard. Also, we may notice that VSR contains the Uncapacitated Swapping Problem, which is also known to be NP-Hard (see [2]).

B. Loading Strategy Flow Model.

Let us suppose now that we are provided with a collection $\Gamma_{Route} = \{\Gamma_{Route,1}, \ldots, \Gamma_{Route,K}\}$ of $K$ carrier routes, all with length $\leq T$.Max. We define a network $H(\Gamma_{Route^*})$ as follows:

- Nodes of $H(\Gamma_{Route^*})$ are:
  - copies of the nodes $x_j^i$ of $\Gamma_{Route,1}, \ldots, \Gamma_{Route,K}$ considered as being all distinct;
  - a source $s$ and a pit $p$;
  - nodes $\text{Exc}(x), x \in X$, excess nodes;
  - nodes $\text{Def}(x), x \in X$, deficit nodes.

- Arrows of $H(\Gamma_{Route^*})$ and related costs $C_e$ are:
  - $\text{tour-arcs} e = (x_j^i, x_{j+1}^i)$ of the routes $\Gamma_{Route,k}$, with cost $C_e = \text{DIST}_{x_j^i, x_{j+1}^i}$;
  - $\text{load-arcs} e = (\text{Exc}(x), x_j^i), x \in X$, x excess, such that the image in $X$ of $x_j^i$ is $x$, with $C_e = 0$;
  - $\text{unload-arcs} e = (y_j^i, \text{Def}(y)), y$ deficit, such that...
the image in X of y, is y, with C*e = 0; excess-arcs e = (s, Exc(x)), x excess, with C_e = 0; deficit-arcs e = (Def(y), p), y deficit, with C_e = 0.

Then we set:

**Load-VSR Model:** {Compute on the network \( H(\Gamma_{\text{Route}}) \) a non negative integral arc indexed flow vector \( Z \) such that:
- for any arc-tour \( e \), \( Z_e \leq \text{CAP} \);
- for any arc \( e = (s, \text{Exc}(x)) \), x excess, \( Z_e = v(x) \);
- for any arc \( e = (\text{Def}(y), p) \), y deficit, \( Z_e = -v(x) \);
- \( C.Z = 0; \) excess-arcs \( e = (s, \text{Exc}(x)) \), x excess, with \( C.Z_e = 0 \).

**Lemma 0:** Any optimal solution (if it exists) of Load-VSR provides us with an optimal loading strategy related to the route collection \( \Gamma^*_{\text{Route}} \).

**Proof:** Any loading strategy related to the tour collection \( \Gamma \) may be turned into a feasible solution of Load-VSR whose cost is exactly the vehicle riding time: \( \Sigma \Sigma_k \text{DIST}(x(k)_k, x(k)_k)_k). \) Conversely, any flow vector \( Z \) which is a feasible solution of Load-VSR can be interpreted as a loading strategy.

We deduce the following **VSR Route Oriented reformulation:** {Compute a route collection \( \Gamma^*_{\text{Route}} = \{\Gamma_{\text{Route}}, ... \} \), and a feasible solution \( Z \) of the related Load-VSR model, such that: \( \alpha.K + \delta.C.Z + \beta.\Sigma.L-Cost(\Gamma_{\text{Route}},k) \) is the smallest possible}.

### III. VSR LOWER BOUNDS

We propose here 2 classes of VSR lower bounds: the first one relies on Min-Cost Assignment models which separately bound the active carrier number \( K \), the carrier riding cost \( \Sigma_k L-Cost(\Gamma(k)) \), and the vehicle riding time \( \Sigma_j \Sigma_k \text{DIST}(x(k)_j, x(k)_j)_j). \) The second one, more complex, embraces the 3 quantities in a same Network-Flow model.

#### A. Min-Cost Assignment Based Lower Bounds

We set the following ILP models:

**VMC Vehicle-Min-Cost:** {Compute integral vector \( Q = (Q_{x,y} \times \text{excess}, y \text{ deficit stations}) \) 0}, such that:
- for any excess station \( x \), \( \Sigma y \text{deficit} Q_{x,y} = v(x) \);
- for any deficit station \( y \), \( \Sigma x \text{excess} Q_{x,y} = -v(y) \);
- minimize \( \Sigma_Q \text{DIST}_{x,y} \).

We denote by LB-VMC the related optimal value, which may be computed while relaxing the integrality constraint on the vector \( Q \).

**UCMC Unit-Carrier-Min-Cost:** {Compute rational vector \( R = (R_{x,y} \times y \text{ stations}) \geq 0 \), such that:
- for any excess station \( x \), \( \Sigma y \text{deficit} R_{x,y} = v(x) \);
- for any deficit station \( y \), \( \Sigma x \text{excess} R_{x,y} = -v(y) \);
- for any subset \( A \) which is not empty and does not contain Depot, \( \Sigma x \in A \), \( y \in A \) \( R_{x,y} \geq 1 \) (No Subtour Constraint).

**LB-UCMC** is the related optimal value.

**CMC Carrier-Min-Cost:** {Compute rational vector \( R = (R_{x,y} \times y \text{ stations}) \geq 0 \), such that:
- for any excess station \( x \), \( \Sigma y \text{deficit} R_{x,y} = \text{CAP}.\Sigma y \text{deficit} R_{x,y} \geq v(x) \);
- for any deficit station \( y \), \( \Sigma x \text{excess} R_{x,y} = \text{CAP}.\Sigma x \text{excess} R_{x,y} \geq -v(y) \);
- for any subset \( A \) which is not empty and does not contain Depot, \( \Sigma x \in A \), \( y \in A \) \( R_{x,y} \geq 1 \) (No Subtour Constraint).

**LB-CMC** is the related optimal value. We denote by LB-Time-CMC the value of the model which derives from CMC by replacing COST by DIST.

**Theorem 1:** LB-MC = \[ \alpha (\text{LB-Time-CMC}/T-Max) + \beta \text{LB-CMC} + \delta \text{LB-VMC} \] is a VSR lower bound.

**Proof:** We see that VLB-A is a lower bound for the vehicle riding cost: \( \Sigma \Sigma_k \text{DIST}(x(k)_k, x(k)_k)_k). \) Also LB-CMC is clearly a lower bound for the carrier riding cost \( \Sigma k \text{Dist}(R_{k}(k)_k) \). We conclude by noticing that the number of carriers \( K \) must be at least equal to the quantity (Carrier Riding Time/T-Max).

**Theorem 2:** A Non Preemptive VSR lower bound is given by LB-UMC = \[ \alpha (\text{LB-Time-UMC}/(\text{CAP}/T-Max)) + \beta \text{LB-UMC}/\text{CAP} + \delta \text{LB-VMC} \] is a VSR lower bound.

**Proof:** We notice that any tour \( \gamma \) which satisfies (E1, E2, E3) may be split into \( \text{CAP tours} | \gamma_1, ... | \gamma_{\text{CAP}} \), with same lengths, which globally perform the relocation process when related \( \text{CAP} = 1 \). So, if Carrier-Ride-Time, and Carrier-Ride-Cost, respectively denote the smallest possible values for the carrier riding time and the carrier riding cost related to the case when \( \text{CAP} = 1 \) and T-Max = \( \infty \), we see that: the Riding Time (Riding Cost) of any VSR solution \( \Gamma \) is at least equal to Carrier-Ride-Time,\( \text{CAP} \) (Carrier-Ride-Cost,\( \text{CAP} \)). We deduce that \( \alpha (\text{Carrier-Ride-Time,\( \text{CAP}/T-Max})) + \beta \text{Carrier-Ride-Cost,\( \text{CAP} + \delta \text{LB-VMC} \) is a VSR lower bound. But Carrier-Ride-
Time$_1$ corresponds to a kind of TSP carrier tour starting and ending from depot, according to which the carrier alternatively moves from excess to deficit nodes. Clearly LB-Time-UCMCA provides us with a lower bound for the DIST-length of such a tour. The same reasoning holds with Carrier-Cost-Time$_1$. We conclude. □

B. A Network Flow Based Lower Bound.

One may reinforce the above lower bounds according to the following construction. We first define a network $G^* = (X^*, E^*)$ as follows:

- $X^* = X \cup \{s, p\}$ where nodes $s$ and $p$ are additional nodes source and pit;
- The restriction of $G^*$ to $X$ is a complete network:
  - any related arc $e = (x, y)$ is provided with a carrier cost $CC_e = COST_{x,y}(\beta + \alpha T_{Max})$ and to a vehicle cost $CV_e = \delta.DIST_{x,y}$.
  - There is an arc $(s, x)$ from $s$ to any excess station $x$, with null carrier and vehicle costs;
  - There is an arc $(y, p)$ from any deficit station $y$ to $p$, with null carrier and vehicle costs;
  - There is a backward arc $(p, s)$, with null carrier and vehicle costs.

![Figure 1](image.png)

Figure 1: A network $G^*$ derived from 3 excess stations and 5 deficit stations.

Then we set:

VSR-Flow Model: (Compute on the network $G^*$ two integral flow vectors $F$ and $f$ such that:
- For any arc $e = ((x, y), x, y \neq x, p$, $f_e \leq \text{CAP}_{F_e}$, $f_e \leq \text{CAP}_{f_e}$
- For any excess (or neutral) station $x$, $f_{i(x, y)} = v(x)$
- For any deficit stations $y$, $f_{i(y, p)} = -v(x)$
- $\sum_{y \in \text{Depot}} F_{\text{Depot}, y} = \sum_{y \in \text{Depot}} F_{\text{Depot}, y} \geq 1$
- Minimize $\sum_{e \in \text{Arcs}} CC_e.F_e + \sum_{e \in \text{Arcs}} CV_e.f_e$

We denote by LB-Flow the related optimal value of this program. Then we may state:

Theorem 3: LB-Flow is a VSR lower bound.

Proof: Any VSR feasible solution may be represented as a tour collection $\Gamma^*$ (it is enough to consider the related route collection $\Gamma^{*}_{\text{route}}$) given together with a feasible solution $Z$ of the linear program Load-VSR. Clearly, $\Gamma$ gives rise to a flow vector $F$. By the same way, $Z$ may be turned into a flow vector $f$ which satisfies (E8), and one easily checks that (E7) is satisfied by the two projections of $\Gamma$ and $Z$ as flow vectors $F, f$ on the network $G^*$. It comes that any VSR feasible solution $\Gamma, Z$ may be turned into a feasible solution $(F, f)$ of VSR-Flow. But the cost of $(\Gamma, Z)$ is equal to: $\alpha.K + \beta.\sum_k L-COST(\Gamma_k) + C.Z$, where $C$ is the cost vector of the Load-VSR model. Proceeding as in the proof of Theorem 1, we see that this quantity is at least equal to: $\alpha.(\sum_k L.DIST(\Gamma_k)/T_{Max}) + \beta.\sum_k L-COST(\Gamma_k) + C.Z$, which coincides with the quantity $CC.F + CV.f$. We conclude. □

Remark 2: The above VSR-Flow model does not solve our VSR problem. On may consider as example, a station set $X = \{\text{Depot, A, B, C}\}$, a carrier flow $F$ which describes the route (Depot, A, B, C, A, Depot) followed by 1 carrier with capacity 1, and a vehicle flow $f$ which routes 1 flow unit from excess station $C$ to deficit station $B$. Then the carrier cannot deliver its load in $B$ before picking it up in $C$.

Remark 3: LB-Flow value provides us with a better lower bound than the LB-MC lower bound of Theorem 3. Still, VSR-Flow is a complex NP-Hard model, whose rational relaxation yields a poor lower bound as soon as CAP is large. The Lagrangean relaxation of the coupling constraint (E7) yields a Lagrangean value $\sup_{\lambda \in \Lambda} (\lambda (CC.(CV + \lambda).h) + H(CC - \lambda).H)$ where:

- Vector flow $h$ is subject to (E8) and vector flow $H$ is subject to (E9),
- $\Lambda = \{\lambda$ such that the restriction of the graph $G_{\text{Proj}}$ to $X$ does not contain any negative $(CC - \lambda)$-circuit$. But, because of total unimodularity of flow constraint matrices, this value is the same as the value obtained by performing Lagrangean relaxation of (E7) on the rational relaxation of VSR-Flow. That means that the above Lagrangean value does not improve the standard relaxation of the integrality constraint.

IV. VSR HEURISTICS

A. Min-Cost Assignment Based Heuristic.

We decompose here the VSR Problem into a Master Min-Cost Assignment problem and a Slave Pick&Delivery (PDP) Problem. Let us recall that a Pick&Delivery instance (see [1, 3, 15]) is defined by:

- a set $J$ of requests $j = (o(j), d(j), \lambda(j))$, where $o(j), d(j)$ and $\lambda(j)$ are respectively the origin, the destination and the load of $j$;
- a maximal duration $D_{Max}$ of the routes followed by the trucks, all with capacity $CH$;
- a Depot node, where all trucks are initially
located;
- a distance matrix \( D \) defined on the set \( N \) of all nodes \( o(j), d(j), j \in J \), augmented with the Depot node.

A collection \( \rho \) of truck routes \( \rho(m), m = 1..M \) defined on the set \( N \) is a feasible PDP solution if:
- Every request \( j \) is serviced by some truck \( m: m \)
  - first loads \( \lambda(j) \) into \( o(j) \) and unloads it in \( d(j) \);
  - The load of a truck never exceeds capacity \( CH \);
  - The length (for the \( D \) matrix) of any route \( \rho(m), m = 1..M \), never exceeds \( D-Max \).

The length, in the sense of the \( D \) matrix, of route \( \rho(m) \), is denoted by \( L-D(\rho(m)) \).

One easily checks that it is possible to impose the following route collection \( \rho \) which minimizes a quantity:
\[
A.M + B. \sum_m L-D(\rho(m)) + C. \sum_j \lambda(j).\text{Ride}(j),
\]
where \( \text{Ride}(j) \) is the time spent by load \( \lambda(j) \) inside a truck. A Load-Split PDP instance is defined the same way, but loads \( \lambda(j) \) may be split into several sub-loads, which are separately handled.

Let us come back to our VSR instance, and suppose that we know, for every pair of stations \( (x, y) \), where \( x \) is an excess station and \( y \) is a deficit station, which quantity \( Q_{xy} \) had to move from \( x \) to \( y \) in order to achieve the Relocation process. Then, we only need to solve the Load-Split PDP instance defined by:
- Requests \( j = (o(j)=x, d(j)=y, \lambda(j)=Q_{xy}) \), taken for all pairs \( x, y \) such that \( Q_{xy} \neq 0 \);
- \( D-Max = T-Max; D = DIST; CH = CAP \);
- \( A = \alpha, B = b, C = \delta \).

One easily checks that it is possible to impose assignment vector \( Q \) to be an optimal solution, for some cost vector \( U = (U_{xy}, x \text{ Excess}, y \text{ Deficit}) \geq 0 \), of the following MCA(U) (Min-Cost Assignment) model:

\[
\text{MCA}(U): \{ \text{Compute integral vector } Q = (Q_{xy}, x \text{ excess, y deficit stations}) \geq 0, \text{ such that:} \}
\]
- For any excess station \( x \), \( \sum_y \text{Deficit} Q_{xy} = v(x) \);
- For any deficit station \( y \), \( \sum_x \text{Excess} Q_{xy} = -v(y) \);
- Minimize \( \sum_{x,y} U_{xy} Q_{xy} \).

This yields the following decomposition scheme VSR-MCA for the handling of the VSR Problem:

**VSR-MCA(N-Rep: Replication Number, N: Loop Number)**

For \( j = 1..N-Rep \) do
- Initialize cost vector \( U = (U_{xy}, x \text{ Excess, y Deficit}) \geq 0 \);
- For \( j = 1..N \) do (*Local Search Loop*)
  - Derive a PDP Assignment vector \( Q \) through optimal resolution of MCA(U);
  - Solve (in a heuristic way) the related Load-Split PDP instance;
  - Update cost vector \( U \);

Apply to the resulting route collection \( \Gamma_\text{Route}^* = \{ \Gamma_\text{Route}(1), \ldots, \Gamma_\text{Route}(K) \} \) the Load-NP-VSR model, and remove from the routes \( \Gamma_\text{Route}(k) \) all stations which do not correspond to any effective load/unload transaction;

Keep the best result ever obtained.

We deal with Load-Split PDP through a GRASP-VNS (Variable Neighborhood Search) process based upon Insert/Remove operators:
- **Insert operator:** Inserting request \( j = (o(j), d(j), \lambda(j)) \) into some route \( \rho(m) \) means:
  - computing 2 insertion nodes \( x \) and \( y \) in \( \rho(m) \), and some sub-load \( \lambda \leq \lambda(j) \);
  - inserting \( o(j) (d(j)) \) between \( x \) (\( y \)) and its successor in \( \rho(m) \);
  - adding \( \lambda \) to the current load of \( \rho(m) \) between \( x \) and \( y \), and updating \( \lambda(j) \);
- **Remove operator:** Delete \( o(j) \) and \( d(j) \) from \( \rho(m) \) and update the load of \( m \) accordingly.

**Cost vector U initialization:** Because of Theorem 1 about LB-MC lower bound, we initialize \( U \) according to the Shortest Cost/Distance Strategy, that means by setting, for any \( x, y, x \text{ Excess, y Deficit, } U_{xy} = \text{DIST}_{xy} + \lambda \cdot (\text{COST}_{xy} + \text{COST}_{xy}) \) where \( \lambda \) is some randomly generated non negative coefficient.

**“Update cost vector U” instruction:**
Let us denote by \( U^0 \) the initial cost vector and let us consider that we are provided with a current cost vector \( U \). We derive from \( U \) a request vector \( Q \), a request set \( \text{Req}(U) = \{ r = (x, y, Q_{xy}) \text{ such that } Q_{xy} \neq 0 \} \) and a VSR feasible solution \( \Gamma^* \), whose global cost \( \text{Global-Cost}(\Gamma^*) \) may be distributed among requests \( (x, y, Q_{xy}) \) in the following way:
- The carrier cost \( \alpha + \beta \cdot \text{COST}(\Gamma(k)) \) of a given carrier \( k \) is shared between the requests which are served by this carrier, proportionally to the value \( \text{L-COST}(\Gamma(k)_{xy}), Q_{xy} \), where \( \Gamma(k)_{xy} \) is the sub-route which is induced by the restriction \( \Gamma(k)_{xy} \) of \( \Gamma(k) \) between \( x \) and \( y \) (in case \( Q_{xy} \) is split into sub-loads, we deal separately with those sub-loads);
  - Every request \( r = (x, y, Q_{xy}) \) is assigned its part \( \text{L-DIST}(\Gamma(k)_{xy}), Q_{xy} \) of the vehicle riding time. It comes that \( \text{Global-Cost}(\Gamma^*) \) may be written \( \text{Global-Cost}(\Gamma^*) = \sum_{r \in \text{Req}(U)} \text{Partial-Cost}(r, \Gamma^*) \), where \( \text{Partial-Cost}(r, \Gamma^*) \) is the part of \( \text{Global-Cost}(\Gamma^*) \) which is charged this way to request \( r \). Then, for every request \( r = (x, y, Q_{xy}) \neq 0 \) we set \( V_{xy} = \text{Partial-Cost}(r, \Gamma^*) Q_{xy} \) and update \( U \) as follows:
  - If \( Q_{xy} \neq 0 \), \( U_{xy} \) is replaced by \( U_{xy} + V_{xy} \)
  - else \( U_{xy} \) is modified;
  - When \( U = U^0 \), \( U \) values may be very different
from V values. So we compute the mean value \( \tau \) of the ratio \( V_{i,j} / U_{i,j} \), \( x,y \) such that \( U_{i,j} \neq 0 \), and replace every value \( U_{i,j}^0 \) by \( \tau U_{i,j}^0 \).

A natural question comes about the quality of the Shortest Distance strategy. We may state:

**Shortest Cost/Distance Theorem 4:** If \( \alpha = \delta = 0 \) (carrier riding time minimization) and \( T - \text{Max} = + \infty \), then the Shortest Cost/Distance Strategy induces an approximation ratio of \( (1 + \text{CAP}) \). This is the best possible ratio.

**Sketch of the Proof.** We first notice that we may,

since \( T - \text{Max} = + \infty \), deal with only one carrier. In order to check that there is no ratio better than \((1+\text{CAP})\), we build a VSR instance as follows:

- \( K = 1 \);
- \( X = \{ \text{Depot} \} \cup \{ \text{o}_{s,c}, \text{d}_{s,c}, n = 0..N-1, c = 1..\text{CAP} - 1 \} \) where \( N \) is a large number; function \( v \) is equal to 1 for \( o_{s,c} \) stations and to -1 for \( d_{s,c} \) stations.
- \( X \) is the node set of a graph \( G = (X, E) \) whose arc set \( E = E_1 \cup E_2 \cup E_3 \cup E_4 \) comes as follows:
  - \( E_1 = \{ (\text{Depot}, \text{o}_{s,1}), (\text{o}_{N-1,1}, \text{Depot}) \} \), both arcs with length equal to \( \frac{1}{2} \);
  - \( E_2 = \{ (\text{o}_{s,c}, \text{o}_{s,c+1}), (\text{d}_{s,c+1}, \text{d}_{s,c}), n = 0..N-1, c = 1..\text{CAP} - 1 \} \), all arcs with length \( e \), where \( e \) is a small number;
  - \( E_3 = \{ (\text{o}_{s,\text{CAP}}, \text{d}_{s,\text{CAP}}), n = 0..N-1 \} \)
  - \( \cup \{ (\text{d}_{s,1}, \text{o}_{s+1,1}), n = 0..N-2 \} \), all arcs with length 1;
  - \( E_4 = \{ (\text{o}_{s,c}, \text{d}_{s,c+\text{CAP}-1}, c = 1..\text{CAP}) \} \) addition being performed modulo \( N \), all arcs with length 1 - \( \alpha \), where \( \alpha \) is a small number.

Then we see that the length of a carrier tour is equal to \( 2n.(1+ (\text{CAP}-1)e) \). But the vector \( Q \) which derives from the Shortest Cost/Distance Strategy Distance strategy is provided by \( E_4 \) and the length of a related optimal PDP solution is equal to \( 2n.(1+ (\text{CAP}-1)e) + \sum_e (2 - \alpha + 2(\text{C}-1)e) \).

In order to check that \((1+\text{CAP})\) provides us with an approximation ratio, we denote by \( Q^{\text{opt}} \) some vector \( Q \) which implements the Shortest Cost/Distance Strategy, and prove that it is possible to derive, from any VSR solution (with only 1 carrier) \( \Gamma \), another feasible solution \( \Gamma^* \) consistent with the Shortest Cost/Distance Strategy. We do in such a way, while using matching techniques, that \( \text{Length}(\Gamma) \leq \text{CAP.Length}(\Gamma) \), and we conclude. \( \square \)

**B. A Vehicle Flow Based Heuristic.**

We derive from the VSR flow model the following heuristic scheme:

**Vehicle-Flow Algorithm.**

**Route collection** \( \Gamma_{\text{Route}} \rightarrow \text{Nil} \);

While coefficients \( v(x), x \in X \) are not null do

Compute an optimal solution \((F^*, f^*)\) of the VSR-Flow model;

**Turn** \( F^* \) into an Eulerian route \( \gamma \);

Split \( \gamma \) into VSR feasible sub-routes \( \gamma_1, \ldots, \gamma_l \), and insert them into \( \Gamma_{\text{Route}}^* \);

Apply the Load-VSR flow model with feasibility oriented objective function: “Maximize \( Z_{p,q} \)”, in order to minimize residual excesses and deficits;

Accordingly update coefficients \( v(x), x \in X \);

Apply to the resulting collection \( \Gamma_{\text{Route}}^* \) the Load-VSR flow model and derive related tour collection \( \Gamma^* \).

We must detail some instructions inside this algorithmic scheme:

- (I1): Handling of the VSR-Flow model: Since VSR-Flow is difficult to handle, we use an ILP library and impose a threshold on the CPU-Time allowed for LB-Flow computation as soon as the number of stations exceeds 30.

- (I2): Deriving an Eulerian route from \( F^* \): Flow vector \( F^* \) defines a collection of arcs \((x, y)\), each of them taken \( F^*_{(x,y)} \) times, in such a way that for any node \( x \), there exists as many arcs which enter into \( x \) as arcs which come out \( x \). So, every connected component \( X_j, j = 1..s \), of the resulting graph gives rise to some Eulerian route \( \gamma_j \). We build \( \gamma \) by starting from \( \text{Depot} \), reaching the closest \( X_j \) into some node \( x_j \), running \( \gamma_j \) until being back to \( x_j \) and then reaching the next closest \( X_j \) and so on. As a matter of fact, since there exists several ways to perform this route construction process, we do it while simulating related loading/unloading transactions and trying to maximize them.

- (I3): Splitting the tour \( \gamma \) into feasible sub-tours: Since \( L - \text{DIST}(\gamma) \) may exceed \( T - \text{Max} \), we run along \( \gamma \) (starting from \( \text{Depot} \)), and every time we arrive to some station \( x \) such that:
  - interrupting current sub-route \( \gamma_j \) by going from \( x \) to \( \text{Depot} \) maintains the feasibility of \( \gamma_j \);
  - going to the successor \( y \) of \( x \) according to \( \gamma \) and next to \( \text{Depot} \) makes \( L - \text{DIST}(\gamma_y) \) exceed \( T - \text{Max} \),

then we close \( \gamma_j \) by going from \( x \) to \( \text{Depot} \), and start \( \gamma_{j+1} \) by going from \( \text{Depot} \) to \( y \) and so on.
V. NUMERICAL EXPERIMENTS

Our purpose is to get a comparative evaluation of both the lower bounds which were described in Section III and the two heuristic scheme described in Section IV, and at testing the influence of scaling coefficients $\alpha, \beta, \delta$. Algorithms were implemented in C, on PC AMD Opteron 2.1GHz, while using gcc 4.1 compiler. We used the CPLEX12 library for the handling of linear models.

Instances: No standardized benchmarks exist for generic VSR. So we built instances as follows:
- The station set $X$ is randomly generated as set of $n + 1$ points $x_0, x_1, \ldots, x_n$ inside the $[0,10] \times [0,10]$ sub-square of the Euclidean 2D-space;
- DIST corresponds to the Euclidean Distance;
- COST corresponds either to a multiple of either the Euclidean distance or the Sum distance $DIST$: $COST(x,y)=|x-x'|+|y-y'|$;
- Each station but Depot $=x_0$ is assigned a random $v(x)$ value chosen between -10 and 10, in such a way that the sum of demands over all stations equal to 0;
- $T_{Max}$ is randomly chosen between 30 and 100.

A. Testing the Impact of Scaling Coefficients $\alpha, \beta, \delta$

On a given instance $(X, v, CAP, T_{Max}, DIST)$, we fix $\alpha = 10$, make vary $\beta, \delta$ with $\beta+\delta = 1$, and compute solutions through the Shortest Distance Strategy. We obtain the Pareto frontier of figure 2, with $t$ is denoting the carrier riding cost and $tInd$ the vehicle riding time.

![Figure 2: Pareto frontier carrier riding cost versus vehicle riding time](image)

Comment: Carrier riding cost and vehicle riding time behave like antagonistic criteria.

B. Comparing the Lower Bounds of Section III

For several groups of 10 instances each related to a given size $n$, we compute the mean value of:
- $LB$ as defined in Theorem 3;
- $LB-MC$ and $LB-UMC$ as defined in Theorem 1 and 2.

We get the following results:

<table>
<thead>
<tr>
<th>$n$</th>
<th>$LB$</th>
<th>$LB-MC$</th>
<th>$LB-UMC$</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>84.8</td>
<td>92.3</td>
<td>73.6</td>
</tr>
<tr>
<td>30</td>
<td>96.5</td>
<td>84.6</td>
<td>77.2</td>
</tr>
<tr>
<td>40</td>
<td>108.4</td>
<td>92.2</td>
<td>89.7</td>
</tr>
<tr>
<td>50</td>
<td>135.1</td>
<td>117.8</td>
<td>112.7</td>
</tr>
<tr>
<td>60</td>
<td>141.5</td>
<td>130.1</td>
<td>115.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$n$</th>
<th>$LB$</th>
<th>$LB-MC$</th>
<th>$LB-UMC$</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>182.7</td>
<td>176.9</td>
<td>160.0</td>
</tr>
<tr>
<td>30</td>
<td>228.2</td>
<td>216.2</td>
<td>210.0</td>
</tr>
<tr>
<td>40</td>
<td>235.6</td>
<td>218.7</td>
<td>205.9</td>
</tr>
<tr>
<td>50</td>
<td>299.9</td>
<td>288.3</td>
<td>269.7</td>
</tr>
<tr>
<td>60</td>
<td>297.3</td>
<td>270.1</td>
<td>261.4</td>
</tr>
</tbody>
</table>

Comment: Experiments confirm Theory (Theorem 3). We notice the quality of the lower bound $LB-LF$.

C. Testing the Heuristics of Section IV

We compute, for the same groups of 10 instances as above, the average of the following Cost values:
- SD (SD(50)) obtained through 1 (50) replications ($N-Rep = 1$ and $N-Rep = 50$) of Shortest Cost/Distance Strategy initialization of $VSR-MCA \Rightarrow CPU-SD$ is the related CPU time (s).
- LS(50): obtained through 50 iterations ($N = 50$, $N-Rep = 1$) of the Local Search loop of $VSR-MCA$, after initialization through $SD \Rightarrow CPU-LS$ is the related CPU time.
- $VF$: obtained through the Vehicle-Flow heuristic $\Rightarrow CPU-VF$ is the related CPU time.
- $LB$ denotes here the $LB$ lower bound of the previous experiment. For $n = 40$ (50, 60) we force the CPLEX computation to stop after 150 s (150 s, 500 s, 1000 s)

We get:

<table>
<thead>
<tr>
<th>$n$</th>
<th>LB</th>
<th>SD</th>
<th>CPU-SD</th>
<th>SD(50)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>84.8</td>
<td>99.5</td>
<td>0.1</td>
<td>94.7</td>
</tr>
<tr>
<td>30</td>
<td>96.5</td>
<td>120.5</td>
<td>0.3</td>
<td>113.6</td>
</tr>
<tr>
<td>40</td>
<td>108.4</td>
<td>152.6</td>
<td>0.9</td>
<td>166.1</td>
</tr>
<tr>
<td>50</td>
<td>135.1</td>
<td>182.3</td>
<td>1.4</td>
<td>169.0</td>
</tr>
<tr>
<td>60</td>
<td>141.5</td>
<td>200.1</td>
<td>1.8</td>
<td>178.5</td>
</tr>
</tbody>
</table>
Comment: The improvement margin induced by the VSR-MCA local loop is not very high, especially when the focus is on the vehicle riding time. A consequence is that performing random diversification according to the Randomized Shortest Cost/Distance Strategy is most often more efficient. Both require small computational time. Conversely, the Vehicle Flow oriented algorithm provides better results but equires higher computation times. At the end, the gap which remains between the LB value and the values which are produced by our heuristics suggests that our lower bound probably misses the optimal value of our VSR problem by about 8%.

II. CONCLUSION

We mainly dealt here with a Vehicle Sharing Relocation problem, related to the operational management of Vehicle Sharing systems, and which we handled according to an approach which puts the focus on the way transported object (vehicles) move from excess stations to deficit ones. Still, many open problems remain, related to the design of exact algorithms, to the way allowing carriers to exchange vehicles may eventually improve the quality of the solutions, and also, if we refer to practical context, to the way algorithms which have been designed for static model may be adapted in order to fit with online contexts. Future research will be carried on in order to address these issues.

### Table 4: Values PI(50), VF with \( \alpha = 10, \beta = 1, \Delta = 0 \)

<table>
<thead>
<tr>
<th>n</th>
<th>LB</th>
<th>LS(50)</th>
<th>CPU-LS</th>
<th>VF</th>
<th>CPU-VF</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>84.8</td>
<td>96.3</td>
<td>5.2</td>
<td>92.3</td>
<td>4.7</td>
</tr>
<tr>
<td>30</td>
<td>96.5</td>
<td>112.5</td>
<td>12.6</td>
<td>108.9</td>
<td>9.6</td>
</tr>
<tr>
<td>40</td>
<td>108.4</td>
<td>139.7</td>
<td>40.4</td>
<td>132.0</td>
<td>140.1</td>
</tr>
<tr>
<td>50</td>
<td>135.1</td>
<td>164.0</td>
<td>61.5</td>
<td>161.8</td>
<td>549.3</td>
</tr>
<tr>
<td>60</td>
<td>141.5</td>
<td>176.7</td>
<td>80.2</td>
<td>169.3</td>
<td>1086.0</td>
</tr>
</tbody>
</table>

### Table 5: Values SD, RSD(50) with \( \alpha = 10, \beta = 0, \Delta = 1 \)

<table>
<thead>
<tr>
<th>n</th>
<th>LB</th>
<th>SD</th>
<th>CPU-SD</th>
<th>RSD(50)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>182.7</td>
<td>220.0</td>
<td>0.1</td>
<td>212.1</td>
</tr>
<tr>
<td>30</td>
<td>228.2</td>
<td>273.1</td>
<td>0.2</td>
<td>264.6</td>
</tr>
<tr>
<td>40</td>
<td>235.6</td>
<td>297.5</td>
<td>0.6</td>
<td>277.7</td>
</tr>
<tr>
<td>50</td>
<td>299.9</td>
<td>372.2</td>
<td>1.0</td>
<td>346.3</td>
</tr>
<tr>
<td>60</td>
<td>297.3</td>
<td>378.4</td>
<td>1.4</td>
<td>348.9</td>
</tr>
</tbody>
</table>

### Table 6: Values PI(50), VF with \( \alpha = 10, \beta = 0, \Delta = 1 \)

<table>
<thead>
<tr>
<th>n</th>
<th>LB</th>
<th>LS(50)</th>
<th>CPU-LS</th>
<th>VF</th>
<th>CPU-VF</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>182.7</td>
<td>217.6</td>
<td>4.1</td>
<td>205.6</td>
<td>5.8</td>
</tr>
<tr>
<td>30</td>
<td>228.2</td>
<td>270.9</td>
<td>8.3</td>
<td>255.7</td>
<td>10.2</td>
</tr>
<tr>
<td>40</td>
<td>235.6</td>
<td>288.7</td>
<td>31.6</td>
<td>264.6</td>
<td>187.1</td>
</tr>
<tr>
<td>50</td>
<td>299.9</td>
<td>364.7</td>
<td>50.7</td>
<td>335.9</td>
<td>561.0</td>
</tr>
<tr>
<td>60</td>
<td>297.3</td>
<td>369.8</td>
<td>61.8</td>
<td>340.8</td>
<td>1098.4</td>
</tr>
</tbody>
</table>

### REFERENCES

15. H. Hernandez Perez, J. Salazar Gonzalez: Heuristics for the on commodity pick up and delivery traveling salesman problem; Transportation Sciences 38, p 244-255, (2004), doi=10.1287/trsc.1030.0086
17. M. Nourinejad, M. J. Roorda: A dynamic carsharing decision support system; Transportation Research E, 66, p 36-50, (2014), DOI: 10.1016/j.tre.2014.03.003