DOI: 10.15439/2017F28
ISSN 2300-5963 ACSIS, Vol. 11

Vehicle Oriented Algorithms for the Relocation of Vehicle Sharing Systems

Alain Quilliot
LIMOS CNRS, Labex IMOBS3
Université Blaise Pascal
63000 Clermont-Ferrand, France
Email: alain.quilliot@isima.fr

Antoine Sarbinowski
LIMOS CNRS, Labex IMOBS3:
Université Blaise Pascal
63000 Clermont-Ferrand, France.

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

- For any $i = 0, .., n(\Gamma)-1$, $T\text{-}Max \geq T_{i+1} \geq T_i + DIST_{xi,xi+1}$; (E1)
- For any $i = 0, .., n(\Gamma)-1$, $0 \leq L^*_i = \Sigma_{j = 0..i} Lj \leq CAP$; (E2)
- $\Sigma_{j = 0..n(\Gamma)} L_j = 0$; (E3)
- For any j such that $v(x_j) \geq 0$, then $v(x_j) \geq Lj \geq 0$; (E4)
- For any j such that $v(x_j) \leq 0$, then $v(x_j) \leq Lj \leq 0$. (E5)

Explanation: (E1): A *carrier* needs at least $DIST_{xi,xi+1}$ time units in order to go from x_i to x_{i+1}; (E2): L^*_i denotes its current load when it leaves x_i, 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 α, β, δ, together with a VSR instance (X, v, *CAP*, *T-Max*, *DIST*), we set:

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

- *For any station x:*
  $\Sigma_k \Sigma_{i \text{ such that } x(k)i = x} L(k)_i = v(x)$. (E6)
- *Minimize* $Cost(\Gamma^*) = \alpha.K + \beta.\Sigma_k L\text{-}Cost(\Gamma(k)) + \delta.(\Sigma_k \Sigma_j (DIST_{x(k)j,\, x(k)j+1}.L^*_j)\}$.

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*: *Cost*(Γ^*) is a weighted sum of the *active carrier* number $\alpha.K$, the *carrier riding cost* Σ_k *L-Cost*($\Gamma(k)$) and the *vehicle riding time* (time *vehicles* spend into the *carriers*) $\Sigma_k \Sigma_j$ ($DIST_{x(k)j,\, x(k)j+1}.L^*_j$).

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 (α very large) with capacity 1, if every quantity v(x) is equal to 1 or -1, and if δ = 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 $\Gamma_{Route}^* = \{\Gamma_{Route,1}, .., \Gamma_{Route,K}\}$ of K *carrier routes*, all with length ≤ *T-Max*. We define a network $H(\Gamma_{Route}^*)$ as follows:

- Nodes of $H(\Gamma_{Route}^*)$ are :
  - ○ copies of the nodes x^k_j of $\Gamma_{Route,1}, .., \Gamma_{Route,K}$, considered as being all distinct;
  - ○ a source s and a pit p;
  - ○ nodes $Exc(x)$, $x \in X$, *excess* nodes;
  - ○ nodes $Def(x)$, $x \in X$, *deficit* nodes.
- Arcs e of $H(\Gamma_{Route}^*)$ and related costs C_e are :
  - ○ *tour-arcs* $e = (x^k_j, x^k_{j+1})$ of the *routes* $\Gamma_{Route,k}$, with cost $C_e = DIST_{\text{xkj, xkj+1}}$;
  - ○ *load-arcs* $e = (Exc(x), x^k_j)$, $x \in X$, x *excess*, such that the image in X of x^k_j is x, with $C_e = 0$;
  - ○ *unload-arcs* $e = (y^k_j, Def(y))$, y *deficit*, such that

the image in X of $y^k{}_j$ 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 $H(\Gamma_{Route}*)$ a non negative integral arc indexed flow vector Z such that:*
○ *for any arc-tour e, $Z_e \leq CAP$;*
○ *for any arc e = $(s, Exc(x))$, x excess, $Z_e = v(x)$;*
○ *for any arc e = $(Def(y), p)$, y deficit, $Z_e = -v(x)$;*
○ *$C.Z = \Sigma_e C_e.Z_e$ is the smallest possible*}

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

Proof: Any *loading strategy* related to the *tour* collection Γ may be turned into a feasible solution of *Load-VSR* whose cost is exactly the *vehicle riding time:* $\Sigma_k \Sigma_j$ $(DIST(x(k)_j, x(k)_{j+1}).L*_j)$. 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 $\Gamma*_{Route} = \{\Gamma_{Route,1}, .., \Gamma_{Route,K}\}$, and a feasible solution Z of the related Load-VSR model, such that: $\alpha.K + \delta.\ C.Z + \beta.\ \Sigma_k\ L\text{-}Cost(\Gamma_{route,k})$ 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* Σ_k *L-COST(Γ(k))* and the *vehicle riding time* $\Sigma_k \Sigma_j$ $(DIST_{x(k)j,\ x(k)j+1}.L*_j)$. 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 $Q = (Q_{x,y}$, x excess, y deficit stations) 0, such that:*
○ *For any excess station x, $\Sigma_{y\ deficit} Q_{x,y} = v(x)$;*
○ *For any deficit station y, $\Sigma_{x\ excess} Q_{x,y} = -v(y)$*
○ *Minimize $\Sigma_{x,y}\ DIST_{x,y}.Q_{x,y}$* }

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 $R = (R_{x,y}, x, y \in X) \geq 0$, such that:*
○ *For any excess station x,*
$\Sigma_{y\ deficit\ station} R_{y,x} = \Sigma_{y\ deficit\ station} R_{x,y} = v(x)$
○ *For any deficit station y,*
$\Sigma_{x\ excess\ station} R_{y,x} = \Sigma_{x\ excess\ station} R_{x,y} = -v(y)$
○ $\Sigma_y\ R_{Depot,y} = \Sigma_y\ R_{y,Depot} \geq 1$
○ *For any subset A, which is not empty and does not contain Depot, $\Sigma_{x \in A\ y,\ y \notin A} R_{x,y} \geq 1$ (No Subtour Constraint)*
○ *Minimize $\Sigma_{x,y}\ COST_{x,y}.R_{x,y}$* }

LB-UCMC is the related optimal value.

CMC Carrier-Min-Cost: {*Compute rational vector $R = (R_{x,y}$, x, y stations$) \geq 0$, such that:*
○ *For any excess station x,*
$CAP.\Sigma_y R_{x,y} = CAP.\Sigma_y\ R_{y,x} \geq v(x)$
○ *For any deficit station y,*
$CAP.\Sigma_x\ R_{x,y} = CAP.\Sigma_x\ R_{y,x} \geq -v(y)$
○ $\Sigma_y\ R_{Depot,y} = \Sigma_y\ R_{y,Depot} \geq 1$
○ *For any subset A, which is not empty and does not contain Depot, $\Sigma_{x \in A\ y,\ y \notin A} R_{x,y} \geq 1$*
○ *Minimize $\Sigma_{x,y}\ COST_{x,y}.R_{x,y},$* }

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* $=\alpha.\lceil LB\text{-}Time\text{-}CMC/T\text{-}Max\rceil + \beta.LB\text{-}CMC + \delta.LB\text{-}VMC$ *is a VSR lower bound.*

Proof: We see that *VLB-A* is a lower bound for the *vehicle riding time:* $\Sigma_k \Sigma_j$ $(DIST_{x(k)j,\ x(k)j+1}.L*_j)$. Also *LB-CMC* is clearly a lower bound for the *carrier riding cost* $\Sigma_k T(k)_{n(\Gamma(k))}$ .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 = $\alpha.\ \lceil LB\text{-}Time\text{-}UCMC/(CAP.T\text{-}Max)\rceil + \beta.LB\text{-}UCMC/CAP + \delta.LB\text{-}VMC$.*

Proof: We notice that any *tour* γ which satisfies (E1, E2, E3) may be split into *CAP tours* $\gamma_1, .., \gamma_{CAP}$, with same lengths, which globally perform the *relocation* process when related *CAP* = 1. So, if *Carrier-Ride-Time*$_1$ and *Carrier-Ride-Cost*$_1$ respectively denote the smallest possible values for the *carrier riding time* and the *carrier riding cost* related to the case when *CAP* = 1 and *T-Max* = + ∞, we see that: the *Riding Time* (*Riding Cost*) of any *VSR* solution Γ is at least equal to *Carrier-Ride-Time*$_1$*/CAP* (*Carrier-Ride-Cost*$_1$*/CAP*). We deduce that $\alpha.\ \lceil Carrier\text{-}Ride\text{-}Time_1/CAP.T\text{-}Max\ \rceil + \beta.\ Carrier\text{-}Ride\text{-}Cost_1/CAP + \delta.LB\text{-}VMC$ 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 $CC_e = COST_{x,y}.(\beta + \alpha/T\text{-}Max)$ 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.

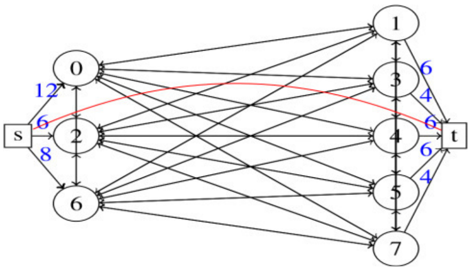

Figure 1: *A network G* derived from 3 excess stations and 5 deficit stations.*

Then we set:

VSR-Flow Model: {*Compute on the network G* two integral flow vectors F and f such that:*

- *For any arc* $e = ((x, y), x, y \neq s, p,$
  $f_e \leq CAP.F_e;$ *(E7)*
- *For any excess (or neutral) station x,*
  $f_{(s,\,x)} = v(x)$ *(E8)*
- *For any deficit deficit station y,*
  $f_{(y\,p)} = -v(x)$ *(E8-1)*
- $\Sigma_y F_{Depot,y} = \Sigma_y F_{y,Depot} \geq 1$ *(E9)*
- *Minimize* $\Sigma_{arcs\ e} CC_e.F_e + \Sigma_{arcs\ e} CV_e.f_e.$}

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 $\Gamma*_{Route}$) given together with a feasible solution Z of the linear program *Load-VSR*. Clearly, Γ 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 Γ and Z as flow vectors F, f on the network G^*. It comes that any *VSR* feasible solution Γ, Z may be turned into a feasible solution (F, f) of *VSR-Flow*. But the cost of (Γ, Z) is equal to: $\alpha.K + \beta.\Sigma_k\ L\text{-}COST(\Gamma_k) + C.Z$, 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: $\alpha.(\Sigma_k\ L.DIST(\Gamma_k))/T\text{-}Max + \beta.\Sigma_k\ L\text{-}COST(\Gamma_k) + C.Z$, 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 $X = \{Depot, A, B, C\}$, 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 $_{\lambda\in\Lambda}$ (Inf $_h$ $(CV + \lambda).h$) + (Inf $_H$ $(CC - \lambda).H$) 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_{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 ρ 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 $\lambda(j)$ in $o(j)$ 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 $D\text{-}Max$.

The length, in the sense of the D matrix, of route $\rho(m)$, is denoted by $L\text{-}D(\rho(m))$. Then solving our *PDP* instance means computing such a feasible *route* collection ρ which minimizes a quantity:

$$A.M + B.\ \Sigma_m\ L\text{-}D(\rho(m)) + C.\ \Sigma_j\ \lambda(j).Ride(j),$$

where $Ride(j)$ is the time spent by load $\lambda(j)$ inside a *truck*. A *Load-Split PDP* instance is defined the same way, but loads λ(j) may be split it 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 $Q_{x,y}$ had to move from x to y *i*n order to achieve the *Relocation* process. Then, we only need to solve the *Load-Split PDP* instance defined by:

- *Requests* $j = (o(j)= x,\ d(j) = y,\ \lambda(j) = Q_{x,y})$, taken for all pairs x, y such that $Q_{x,y} \neq 0$;
- $D\text{-}Max = T\text{-}Max$; $D = DIST$; $CH = CAP$;
- $A = \alpha$, B= b, C = δ.

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

MCA(U):{*Compute integral vector* $Q = (Q_{x,y},\ x$ *excess, y deficit stations*) ≥ 0, *such that:*

- *For any excess station x,* $\Sigma_{y\ deficit}\ Q_{x,y} = v(x)$;
- *For any deficit station y,* $\Sigma_{x\ excess}\ Q_{x,y} = -\ v(y)$;
- *Minimize* $\Sigma_{x,y}\ U_{x,\ y}.Q_{x,y},\ .\}$

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 $j = 1..N\text{-}Rep$ do
  Initialize cost vector $U = (U_{x,y},\ x\ Excess,\ y\ Deficit) \geq 0$;
  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 $\Gamma_{Route}* = \{\Gamma_{Route}(1), .., \Gamma_{Route}(K)\}$ the Load-NP-VSR model, and remove from the routes $\Gamma_{Route}(k)$ 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 $j = (o(j), \mathrm{d}(j), \lambda(j))$ 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 λ 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 *Excess*, y *Deficit*, $U_{x,y} = DIST_{x,y.} + \lambda.\ (COST_{x,y.} + COST_{y,x})$ where λ 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 $Req(U) = \{r = (x, y, Q_{x,y})$ such that $Q_{x,y} \neq 0\}$and a *VSR* feasible solution Γ*, whose global cost $Global\text{-}Cost(\Gamma^*)$ may be distributed among requests $(x, y, Q_{x,y})$ in a natural way:

- The *carrier* cost $\alpha + \beta.L\text{-}COST(\Gamma(k))$ related to a given *carrier* k is shared between the *requests* which are served by this *carrier*, proportionally to the value $L\text{-}COST(\Gamma(k)_{x,y}).Q_{x,y}$, where $\Gamma(k)_{x,y}$ is the sub-route which is induced by the restriction $\Gamma(k)_{x,y}$ of $\Gamma(k)$ between x and y (in case $Q_{x,y}$ is split into sub-loads, we deal separately with those sub-loads);
- Every *request* $r = (x, y, Q_{x,y})$ is assigned its part $L\text{-}DIST(\Gamma(k)_{x,y}).Q_{x,y}$ of the *vehicle riding time.* It comes that $Global\text{-}Cost(\Gamma^*)$ may be written $Global\text{-}Cost(\Gamma^*) = \Sigma_{r \in Req(U)}\ Partial\text{-}Cost(r, \Gamma^*)$, where $Partial\text{-}Cost(r, \Gamma^*)$ is the part of $Global\text{-}Cost(\Gamma^*)$ which is charged this way to request r. Then, for every request $r = (x,y, Q_{x,y} \neq 0)$ we set $V_{x,y} = Partial\text{-}Cost(r, \Gamma^*)\ Q_{x,y}$ and update U as follows:
- If $Q_{x,y} \neq 0$, $U_{x,y}$ is replaced by $(U_{x,y} + V_{x,y})/2$ else $U_{x,y}$ is unmodified;
- When $U = U^0$, U values may be very different

from V values. So we compute the mean value τ of the ratio $V_{x,y} / U_{x,y}$, x,y such that $Q_{x,y} \neq 0$, and replace every value $U^0_{x,y}$ by $= \tau.U^0_{x,y}$.

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-Max = + ∞, then the Shortest Cost/Distance Strategy induces an approximation ratio of (1+CAP). This is the best possible ratio.*

Sketch of the Proof. We first notice that we may, since*T-Max* = + ∞, deal with only one *carrier*. In order to check that there is no ratio better than (1+*CAP*), we build a VSR instance as follows:

- $K = 1$;
- $X = \{Depot\} \cup \{o_{n,c}, d_{n,c}, n = 0..N\text{-}1, c = 1..CAP - 1\}$ where N is a large number; function v is equal to 1 for $o_{n,c}$ stations and to – 1 for $d_{n,c}$, 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:
    - ○ $E_1 = \{(Depot, o_{0,1}), (o_{N\text{-}1,1}, Depot)\}\}$, both arcs with length equal to ½;
    - ○ $E_2 = \{(o_{n,c}, o_{n,c+1}), (d_{n,c+1}, d_{n,c}), n = 0..N\text{-}1, c = 1..CAP - 1\}$, all arcs with length ε, where ε is a small number;
    - ○ $E_3 = \{(o_{n,CAP}, d_{n,CAP}), n = 0..N\text{-}1\}$
      $\cup \{(d_{n,1}, o_{n+1,1}), n = 0..N\text{-}2\}$,
      all arcs with length 1;
    - ○ $E_4 = \{\{(o_{n,c}, d_{n+c\text{-}CAP\text{-}1,c}), \}, n = 0..N\text{-}1, c = 1..CAP\}$ addition being performed modulo N, all arcs with length 1- α, where α is a small number}.

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

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

B. A Vehicle Flow Based Heuristic.

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

***Vehicle-Flow* Algorithm**.

```
Route collection $\Gamma_{Route}^* \leftarrow Nil$;
While coefficients $v(x)$, $x \in X$ are not null do
    Compute an optimal solution $(F^*, f^*)$ of the VSR-Flow model;    (I1)
    Turn $F^*$ into an Eulerian route $\gamma$;    (I2)
    Split $\gamma$ into VSR feasible sub-routes $\gamma_1, .., \gamma_s$ and insert them into $\Gamma_{Route}^*$;    (I3)
    Apply the Load-VSR flow model with feasibility oriented objective function: "Maximize $Z_{p,s}$" in order to minimize residual excesses and deficits;
    Accordingly update coefficients $v(x)$, $x \in X$;
Apply to the resulting collection $\Gamma_{Route}^*$ 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 γ_j. We build γ by starting from *Depot*, reaching the closest X_j into some node x_j, running γ_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 γ into feasible sub-tours*: Since *L-DIST*(γ) may exceed *T-Max*, we run along γ (starting from *Depot*), and every time we arrive to some station x such that:
    - ○ interrupting current sub-route γ_j by going from x to *Depot* maintains the feasibility of γ_j,
    - ○ going to the successor y of x according to γ and next to *Depot* makes *L-DIST*(γ_j) exceed *T-Max,*

  then we close γ_j by going from x to *Depot*, and start γ_{j+1} by going from *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 α, β, δ.

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, .., x_n, inside the [0,10] × [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-S*: $COST_{(x,y),(x',y')} = |x' - x| + |y' - y|$;
- 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 α,β, δ.

On a given instance (*X*, *v*, *CAP*, *T-Max*, *DIST*), we fix α = 10, make vary β, δ with β+δ = 1, and compute solutions through the *Shortest Distance Strategy*. We obtain the Pareto frontier of figure 2, with *ts* denoting the *carrier riding cost* and *tind* the *vehicle riding time*.

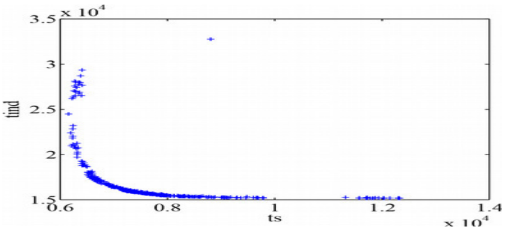

Figure 2: *Pareto frontier carrier riding cost versus vehicle riding time*

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-Flow*: as defined in Theorem 3;
- *LB-MC* and *LB-UMC* as defined in Theorem 1 and 2.

We get the following results:

Table 1: Lower Bounds with A =10, B =1,Δ =0

n	*LB-Flow*	*LB-MC*	*LB-UMC*
20	84.8	82.3	73.6
30	96.5	84.6	77.2
40	108.4	92.2	89.7
50	135.1	117.8	112.7
60	141.5	130.1	115.2

Table 2: Lower Bounds with A =10, B =0,Δ =1

n	*LB-Flow*	*LB-MC*	*LB-UMC*
20	182.7	176.9	160.0
30	228.2	216.2	210.0
40	235.6	218.7	205.9
50	299.9	288.3	269.7
60	297.3	270.1	261.4

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* => *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* => *CPU-LS* is the related CPU time.
- *VF*: obtained through the *Vehicle-Flow* heuristic => *CPU-VF* is the related CPU time.
- *LB* denotes here the *LB-Flow* 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 3: Values SD, RSD(50) with A =10, B =1,Δ =0

n	*LB*	*SD*	*CPU-SD*	*SD(50)*
20	84.8	99.5	0.1	94.7
30	96.5	120.5	0.3	113.6
40	108.4	152.6	0.9	166.1
50	135.1	182.3	1.4	169.0
60	141.5	200.1	1.8	178.5

TABLE 4: VALUES PI(50), VF WITH A =10, B =1,Δ =0

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

TABLE 5: VALUES SD, RSD(50) WITH A =10, B =0,Δ =1

n	LB	SD	CPU-SD	SD(50)
20	182.7	220.0	0.1	212.1
30	228.2	273.1	0.2	264.6
40	235.6	297.5	0.6	277.7
50	299.9	372.2	1.0	346.3
60	297.3	378.4	1.4	348.9

TABLE 6: VALUES PI(50), VF WITH A =10 ,B =0,Δ =1

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

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 *on line* contexts. Future research will be carried on in order to address these issues.

REFERENCES

[1] C. Archetti, M. Speranza: *The split delivery vehicle routing problem, a survey;* in The vehicle routing problem: latest advances and new challenges; p 103-122, Springer U.S, (2008). DOI: 10.1111/j.1475-3995.2011.00811.x

[2] S. Anily, M. Gendreau, G. Laporte: *Uncapacitated swapping problem on line and circle* ; Networks 58, p 83-94, (2011). DOI: 10.1016/j.dam.2012.07.002

[3] B. Bernay, S. Deleplanque, A. Quilliot: Routing in Dynamic Networks: Grasp Versus Genetics, 7 th WCO Workshop, FEDCIS Conf, Warwsaw, p 487, 492, (2014), DOI: http://dx.doi.org/10.15439/978-83-60810-58-3

[4] J. Aronson: A survey on dynamic network flows; Annals of O.R, 20, p 1-66, (1989). DOI: 10.1007/BF02216922

[5] M. Barth, M. Todd: *Simulation model analysis of a multiple station shared vehicle system*. Transp. Res. C, 7(4): 237–259, (1999), DOI: 10.1016/S0968-090X(99)00021-2

[6] M. and P. Benchimol, B. Chappert, A. De la Taille, F. Laroche, F. Meunier, L. Robinet : *Balancing the stations of a self service bike systems*, RAIRO-RO 45, p 37-61, (2011). DOI: http://dx.doi.org/10.1051/ro/2011102

[7] C. Bordenave, M. Gendreau, G. Laporte: *A branch and cut algorithm for the preemptive swapping problem*; Networks 59, 4, p 387-399, (2012). DOI: 10.1002/net.20447

[8] B. Boyaci, K. Zografos, N. Geroliminis: *An optimization framework for the development of efficient one-way car-sharing systems*. EJOR, 240(3):718–733, (2015), DOI: 10.1016/j.ejor.2014.07.020

[9] A. Carlier, A. Munier-Kordon, W. Klaudel: *Mathematical model for the study of relocation strategies in one-way carsharing systems.* Transp. Res. Procedia, 10:374–383, (2015). DOI: 10.1016/j.trpro.2015.09.087

[10] D. Chemla, F. Meunier, R. Wolfler Calvo: *Bike sharing systems: solving the static rebalancing problem*; Discrete Optimization 10 (2), p 120-146, (2013). doi:10.1016/ j.disopt.2012.11.005

[11] C. Contardo, C. Morency, L. M. Rousseau: *Balancing a dynamic public bike-sharing system*; Rapport CIRRELT, Univ. MONTREAL (2012).

[12] J. F. Cordeau, G. Laporte: *A Tabu search heuristic algorithm for the static multi-vehicle Dial and Ride Problem*; Transportation Research B 37, p 579-594, (2003), doi.org/10.1016/S0191-2615(02)00045-0

[13] D. Gavalas, C. Konstantopoulos, G. Pantziou: *Design and management of vehicle sharing systems: a survey of algorithmic approaches*; ArXiv e-prints, arxiv.org/pdf/1510.01158, Oct. 2015. arXiv:1510.01158v1

[14] G. H. Kek, R. L. Cheu, Q. Meng, C. Ha Fung: *A decision support system for vehicle relocation operations in carsharing systems*; Transportation Research E: Logistics and Transportation Review 45 (1), p 149-158, (2009). doi:10.1016/j.tre.2014.01.007

[15] H. Hernandez Perez, J. Salazar Gonzalez: *Heuristics for the one commodity pick up and delivery traveling salesman problem*; Transportation Sciences 38, p 244-255, (2004), doi>10.1287/trsc.1030.0086

[16] J. Lee, G. L. Park: *Design of a team based relocation scheme in electric vehicle sharing systems*; Proc. Int. Conf. Computat. Sci. and Applications 7973, p 368-377, (2013).

[17] M. Nourinedjad, 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

[18] M. Rainer-Harbach, P. Papazeck, B. Hu, G. Raidl: *Balancing bicycle sharing systems: a variable neighbourhood search approach*, Journal of Global optimization, 63-3, p 597-629 (2015)

[19] T. Raviv, M. Tzur, I. Forma: *Static repositioning in a bike sharing system: models and solution approaches*; EURO Journal on Transportation and Logistics 2 (3), p 187-229, (2013).DOI A0.1007/a13676-012-0017-6.