Pesquisa Operacional (2017) 37(3): 455-486
© 2017 Brazilian Operations Research Society
Printed version ISSN 0101-7438 / Online version ISSN 1678-5142
www.scielo.br/pope
doi: 10.1590/0101-7438.2017.037.03.0455

NETWORK FLOW ORIENTED APPROACHES FOR VEHICLE SHARING RELOCATION PROBLEMS

Alain Quilliot[1*], Samuel Deleplanque[2],
Antoine Sarbinowski[3] and Annegret Wagler[4]

Received April 24, 2017 / Accepted November 30, 2017

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 propose and study here several network flow oriented models and algorithms which deal with a static version of this problem while unifying *preemption* and *non preemption* as well as *carrier riding cost*, *vehicle riding time* and *carrier* number minimization. Those network flow models are vehicle driven, which means that they focus on the way vehicles are exchanged between excess and deficit stations. We perform a lower bound and approximation analysis which leads us to the design and test of several heuristics. One of them involves implicit dynamic network handling.

Keywords: Network Flow, routing, Vehicle sharing.

1 INTRODUCTION

Vehicle Sharing systems [14, 20, 28, 29] 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* may be bicycles or electric cars. There exists a special station called *Depot*, in which a set of *carriers* (trucks, self-platoon convoys, ...) are waiting: they periodically exchange *vehicles* between the stations and eventually provide them with additional *vehicles*. A trend is to make the system be a *one-way* system: users are not imposed to give *vehicles* back at the *stations* where they picked them up. This feature makes the system more attractive, but also raises the eventuality of unbalanced situations: stations may become overfilled other under-filled, provoking local shortages or making users unable to give their *vehicles* back. This makes arise two decision problems:

*Corresponding author.
[1]LIMOS CNRS 6158, Labex IMOBS3, UCA Clermont-Ferrand, France. E-mail: alain.quilliot@isima.fr
[2]ULB, Bruxelles, Belgium. E-mail: deleplanque.samuel@gmail.com
[3]Labex IMOBS3, UCA Clermont-Ferrand, France. E-mail: antoine.sarbinowski@isima.fr
[4]LIMOS CNRS, UCA Clermont-Ferrand, France. E-mail: awagler@isima.fr

– a *strategic* level problem [8, 9, 14, 25, 29], about the way stations are located and capacitated and about the pricing of the system [29]. One must simultaneously maximize some global *Access Demand*, and minimize costs which involve not only infrastructure costs but also running costs related to periodical *vehicle* relocation. Though this *Vehicle Sharing Station Location* (VSSL) problem looks like a standard *Facility Location* problem [13, 19, 22, 26, 30], addressing it is difficult in practice, since estimating the way *Access Demand* depends on the way stations are located can only be done through rough approximation.

– an *operational* (or tactical) level problem (see [5, 6, 7, 10, 11, 18, 20, 21, 23, 24, 25, 27]), about the way *vehicles* are periodically moved from *excess* to *deficit* stations in order rebalance the system (*Relocation Process*). Performing this process while meeting both economic and quality of service purposes means addressing a *Vehicle Sharing Relocation* problem (VSR).

This contribution is devoted to the operational level, that means to the VSR Problem, which also appears as a slave sub-problem in any bi-level VSSL formulation. Related VSR models may be:

– *static*: at some time during the process, *excess* and *deficit* stations are identified, together with *excess* and *deficit* amount of *vehicles*. One must make the *carriers* move *vehicles* from *excess* to *deficit* stations, while minimizing some operational cost, function of the *vehicle riding time*, of the number of *carriers* and of the *carrier riding time*, while keeping the total duration of the process from exceeding a *makespan* threshold;

– *dynamic*: one knows, for every station x, at which time *vehicles* are going to be demanded or given back by the users. Then one schedules the *carriers* in order to meet most demands and avoid any unbalanced situation, while minimizing some operational cost;

– *on line*: the context is the same as in the dynamic case, but knowledge about demands is incomplete and uncertain.

Preemption may be allowed: a *carrier* may load some *vehicle* at some station and drop it at another station, before some other (or eventually the same) *carrier* comes, loads it again and brings it until a third station.

In practice, VSR models have to be handled *on line* [4, 18, 20, 23]: relocation is performed in a continuous way and, at any time, knowledge about customer requests is incomplete and uncertain. Still, as it is usual when it comes to scheduling or routing decisional problems, it is appropriated, in order to better understand the problem and design efficient strategies, to first deal with a *static* or eventually a *dynamic* version. VSR literature makes appear several *static* models (see [3, 4, 10]) which have been addressed through *metaheuristic schemes* or through hierarchical decomposition into a routing *master* model, handled through local search, and a load/unload network flow *slave* model [1]. Most authors impose restrictions on the number of *carriers* and the components of the cost function [5, 6, 7, 10], most often reduced to the *carrier riding time*.

Some authors consider time indexed requests [11, 27] and address the resulting model through a Benders decomposition scheme. None of them links *non preemption* and *preemption*, while, even if not practical from the point of view of a central manager, *preemption* may be used as a relaxation of *non preemption* and help into designing algorithms.

More, one may notice that a common feature of the above mentioned static and dynamic models is that they are *carrier* oriented [24, 25], 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 [10]. Such an approach may be criticized because of the lack of a backward link between the master *carrier* tour collection and the *vehicle* sub-problem: the search for the master *carrier* tour collection is then performed in a somewhat blind way (genetic algorithms, ...).

So we adopt here the opposite point of view and consider that performing a *relocation* process means routing *vehicles* from *excess stations* to *deficit* ones in a way which make them share, as often as possible, related *carriers*. This leads us to propose models which stress the role played by the *vehicle* network flow induced by the relocation process, and then derive alternative approaches to *carrier driven* ones, which we say to be *vehicle driven*: the *vehicle* routing strategy becomes the master object, which determines in turn the *carrier* routes. This allows us to link *preemptive* and *non preemptive* VSR models and point out that understanding *preemption* as a relaxation of *non preemption* leads us to a common *Network Flow* framework.

The paper is organized as follows. We first provide (Section 2) a general framework for both *preemptive* and *non preemptive* static VSR, which mixes several performance criteria: economic cost of the *relocation* process (*carrier number* and *carrier riding cost*), and quality of service (unavailability of the *vehicles* during the process). We reformulate resulting models as *Network Flow* models, making appear *preemption* as a relaxation of *non preemption*. We keep on (Section 3) by performing a lower bound analysis of this VSR model. In Section 4, we propose a first heuristic scheme, which considers the way *vehicles* are distributed from *excess stations* to *deficit* ones as the master object of a *Min Cost Assignment/Pick up and Delivery* hierarchical decomposition scheme, and state an approximation result for what we call the *min-cost assignment strategy*. In Section 5 and 6, we propose and test heuristics, which deal with aggregated *vehicle* and *carrier* flow vectors and turn them into solutions of respectively *Non Preemptive* and *Preemptive* VSR. One of those heuristics involves the implicit management of large size dynamic network.

2 VSR PREEMPTIVE AND NON PREEMPTIVE CARRIER ORIENTED 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 *carriers* 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 *carriers* have to remove $v(x)$ *vehicles* from x;

if $v(x) = 0$ then x is said to be *neutral*. *Carriers* are initially located at *Depot* and they all have a same capacity *CAP*. We suppose that $\Sigma_{x \in X} v(x) = 0$, which means that some stations may be used to bring additional *vehicles* to the system, or, conversely, to remove some of them. We also suppose that *Depot* is *neutral*. Any station x is provided with a capacity $C(x)$. *DIST* denotes the *X.X time* matrix: $DIST_{x,y}$ is the time required for a *carrier* to go from station x to station y. *T-Max* is the maximal *makespan* of the *relocation* process: the total time for this process cannot exceed *T-Max*. By the same way, *COST* denotes the *X.X carrier cost* matrix: $COST_{x,y}$ is the integrated cost (energy, human resource, ...) induced by a move of a *carrier* from station x to station y, when this move is performed in $DIST_{x,y}$ time units. Both matrices *DIST* and *COST* satisfy the *Triangle Inequality* and are such that $COST_{x,x} = DIST_{x,x} = 0$ for any station x. *Idle-Cost* denotes the *waiting cost* induced for a *carrier* when it remains at any station $x \neq Depot$ during one time unit. We suppose (*Extended Cost Hypothesis*) that if a *carrier* moves from x to y at a reduced speed in time $t \geq DIST_{x,y}$, then the induced extended cost $E\text{-}COST_{x,y,t}$ is equal to $COST_{x,y} + Idle\text{-}Cost.(t - DIST_{x,y})$. All this defines a *VSR* instance (X, v, C, *CAP*, *T-Max*, *DIST, COST*).

2.1 Non Preemptive VSR Model

A *VSR tour* Γ 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 $\Gamma_{Load} = \{L_0 = 0, L_1, \ldots, L_{n(\Gamma)}\}$ and $\Gamma_{Time} = \{T_0 \geq 0, T_1, \ldots, T_{n(\Gamma)}\}$ of coefficients whose meaning is: a *carrier* which follows the route Γ_{Route} loads, at time T_i, L_i vehicles at station x_i (unloads in case $L_i < 0$). The length of Γ_{Route} in the *COST* sense is given by $L\text{-}COST(\Gamma_{Route})$ $= \Sigma_j COST_{xj,x_j+1}$. The length of Γ_{Route} in the *DIST* sense is given by $L\text{-}DIST(\Gamma_{Route}) = \Sigma_j$ $\text{DIST}_{xj,xj+1}$. The cost $L\text{-}E\text{-}COST(\Gamma)$ of Γ is given by: $L\text{-}E\text{-}COST(\Gamma) = L\text{-}COST(\Gamma_{Route}) + Idle\text{-}Cost.(T_{n(\Gamma)} - T_0)$. For any i, we denote by $L_i^* = \Sigma_{j=0..i} Lj$ the load of the *carrier* when it leaves station x_i.

This *VSR tour* Γ is *Non Preemptive* VSR *feasible* if:

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

Explanation: (E1): A *carrier* needs at least $DIST_{xi,xi+1}$ time units to go from x_i to x_{i+1}; (E2, E3): Current *carrier* load L_i^* cannot exceed the capacity *CAP*, and this load is null at the end of the *tour*; (E4): loading (unloading) operations are respectively restricted to *excess* (*deficit*) stations, which means that we impose a given *vehicle* to be moved from an *origin* station to a *destination* station by exactly one *carrier* (*Non Preemption*).

Given scaling coefficients α, β, δ together with a VSR instance (*X*, *v*, *C*, *CAP*, *T-Max*, *DIST*, *COST*), we set:

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

- *For any station* x: $\Sigma_k \Sigma_{i \text{ such that } x(k)i = x} L(k)_i = v(x)$. (E5)
- *Minimize Global-Cost*$(\Gamma^*) = \alpha.K + \beta.\Sigma_k$ *L-E-COST*$(\Gamma(k))+$ $\delta.(\Sigma_k \Sigma_j (DIST_{x(k)j,x(k)j+1}.L^*(k)_j)\}$.

Explanation: (E5): 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. *Global-Cost*(Γ^*) is a weighted sum of the *carrier number*, the *carrier riding cost*, and the *vehicle riding time* (time *vehicles* spend into the *carriers*).

Remark 1. Because of *non preemption*, every move from $x(k)_i$ to $x(k)_{i+1}$ in $\Gamma(k)$ may be performed in exactly $DIST_{x(k)i\ x(k)i+1}$ time units; Thus Σ_k *L-E-COST*$(\Gamma(k))$ may be replaced by Σ_k *L-COST*$(\Gamma(k)_{Route}$. By the same way, any *neutral* stations but the *Depot* station may be removed from the input of the *Non Preemptive* VSR model.

Remark 2 (About MIP Models and Complexity). Modeling *VSR* through a MIP (*Mixed Integer Linear Program*) is possible, but inefficient. The reason is that there is no a priori bound about the number of times a given station is going to be visited by a same *carrier*. As for complexity, in the case when $K = 1$ (α very large), $v(x)$ values are equal to 1 or -1, $CAP = 1$ and $\delta = 0$, our problem is equivalent to the *Travelling Salesman* Problem set on a bipartite graph (the *excess stations* on one side and the *deficit* ones on the other side), which is NP-Hard. *Non Preemptive VSR* also contains the *Uncapacitated Swapping Problem*, which is also NP-Hard (see [1]).

2.2 Loading Strategy Flow Model Related to a VSR Route Collection Γ^*_{Route}

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*. Following [10], we define a network $H(\Gamma_{Route})$ as follows (see Fig. 1):

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

- Arcs e of $H(\Gamma_{Route})$ and related costs C_e are:
  - *Route arcs* $e = (x(k)_j, x(k)_{j+1})$ of the *routes* $\Gamma_{Route}(k)$, with cost $C_e = DIST_{x(k)j,x(k)j+1}$;
  - *Excess 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 $C_e = 0$;
  - *Deficit arcs* $e = (y(k)_j, Def(y))$, y *deficit*, such that $y(k)_j$ is y, with $C_e = 0$;
  - *Input arcs* $e = (s, Exc(x))$, x *excess*, and *output arcs* $e = (Def(y), p)$, y *deficit*, with $C_e = 0$.

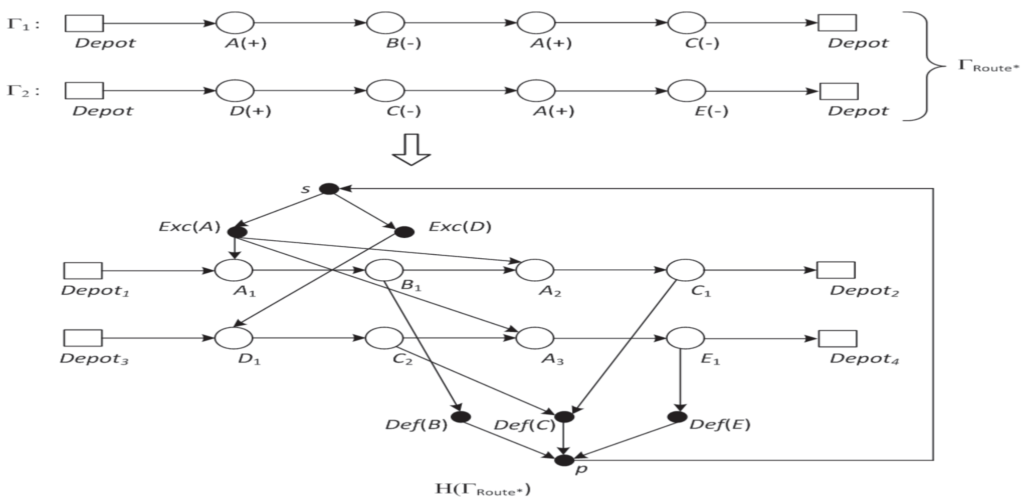

Figure 1

Then we may set:

Load-NP-VSR Model: {*Compute on* $H(\Gamma^*_{Route})$ *a non negative integral arc indexed flow vector* Z *such that*:

- *for any route arc* e, $Z_e \leq CAP$;
- *for any input arc* $e = (s, Exc(x)), xexcess, Z_e = v(x)$;
- *for any output arc* $e = (Def(y), p), ydeficit, Z_e = -v(x)$;
- *Cost* $C.Z = \Sigma_e C_e.Z_e$ *is the smallest possible*}.

This construction yields, as in [10]:

Lemma 0. *Any optimal solution (if it exists) of Load-NP-VSR provides us with an optimal loading strategy related to the route collection* Γ^*_{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:* Σ_k, Σ_j $(DIST(x(k)_j, x(k)_{j+1}).L_j^*)$. Conversely, any flow vector Z which is a feasible solution of *Load-NP-VSR* can be interpreted as a *loading strategy*. □

It comes that *Non Preemptive* VSR may be reformulated:

Non Preemptive VSR Reformulation: {*Compute* $\Gamma^*_{Route} = \{\Gamma_{Route}(1)\ldots, \Gamma_{Route}(K)\}$*, together with an optimal Load-NP-VSR solution* $Z(\Gamma^*_{Route})$*, which minimize:*
$\alpha.K + \delta.C.Z(\Gamma^*_{Route}) + \beta.\Sigma_k L - COST(\Gamma_k)\}$.

2.3 Preemptive VSR Model

In case preemption is allowed, then we say that the *VSR tour* Γ *is preemptive VSR feasible* if (E1, E2, E3) are true. Besides, for any collection $\Gamma^* = (\Gamma(k), k = 1..K \leq K\text{-}Max)$ of such non-preemptive feasible tours, we set, for any time value t, and any station x:

- $\Delta(\Gamma, x, t) = \{(k, j), k = 1\ldots K, j = 0\ldots n(\Gamma(k))$, such that $x(k)_j = x$ and $T(k)_j \leq t\}$;
- $H(\Gamma, x, t) = \text{Sup}(0, v(x)) - \Sigma_{(k,j)\in\Delta(\Gamma,x,t)} L(k)_j$.

Clearly, $H(\Gamma, x, t)$ denotes the number of *vehicles* which are really located in station x at time t after all loading/unloading transactions have been performed. Then we say that the collection $\Gamma = (\Gamma(k), k = 1\ldots K \leq K - Max)$ is a *feasible* solution for the *preemptive VSR* instance (*X*, *v*, *C*, *CAP*, *T-Max*, *DIST, COST*) if every $\Gamma(k)$ is *preemptive feasible*, if (E5) holds and if, for any time value t and any station x: $0 \leq H(\Gamma, x, t) \leq C(x)$. (E6)

(E6) expresses the fact that, at any time t, the number of *vehicles* currently located at x is non negative and cannot exceed the capacity of the station x. Then we may set:

Preemptive VSR Model: {*Compute a preemptive VSR feasible tour collection* $\Gamma^* = (\Gamma(k), k = 1\ldots K)$ *such that (E1, E2, E3, E5 and E6 hold) and which minimizes the following global cost:*

- *Global-Cost*$(\Gamma^*) = \alpha.K + \beta.\Sigma_k$ *L-E-COST*$(\Gamma(k))$
  $+\delta.(\Sigma_k \Sigma_j(DIST_{x(k)j,x(k)j+1}.L^*(k)_j))\}$.

Remark 3. The use of *preemption* leads to the introduction of synchronization mechanisms. A carrier k which arrives at some station x may wait for another vehicle k' before leaving x. So *L-E-COST* cannot any more be replaced by *L-COST* in above *Global-Cost*. The role of the *Extended Cost Hypothesis* is that there is no difference, from the *Global-Cost* point of view, between moving from some station x until some station y according to a maximal speed strategy and next waiting some time t at y, and moving from x to y at a reduced speed in order to arrive in y with a delay t. Also, the *vehicle riding time* $\Sigma_k \Sigma_j$ $(\text{DIST}_{x(k)j,x(k)j+1}.L_j^*)$ quantity expresses the time *vehicles* spend into the *carriers*: in case some *carrier* k arrives to some station x at time

T and leaves it at time $T+t$, *vehicles* unloaded at time t and loaded again at time $T+t$ (provided $C(x)$ is large enough) are not involves in this *vehicle riding time* since they are available for users between T and $T+t$.

Remark 4. Taken together, above *Non Preemptive* and *Preemptive* VSR models extend [5, 7, 10, 11, 14, 25], since they unify *preemption* and *non preemption*, and mix *carrier numbers, vehicle riding time* and *carrier riding cost* into a same criterion. Still, in case of non feasibility, we do not take into account, as in [10, 11], the eventual deviation between the wanted balanced state and the true state of the system at the end of the process.

2.4 A Network Flow Framework

Let us recall that a *flow vector* defined on a network $G = (N, A)$, with node set N and arc set A, is a rational (or integral) valued A-indexed vector g such that, for any node z, the following *flow conservation law* holds:

$$\Sigma_{e \text{ such that } origin(e) = z} g_e = \Sigma_{e \text{ such that } destination(e) = z} g_e$$

Let us now suppose that all values $DIST_{x,y}$ are integral (it is always possible to do it). Then we derive from the VSR instance (*X*, *v*, *C*, *CAP*, *T-Max*, *DIST, COST*) a *dynamic network* [4] $G_{T\text{-}Max} = (X_{T\text{-}Max}, E_{T\text{-}Max})$ as follows (see Fig. 2):

- $X_{T\text{-}Max}$ is the set of all pairs (x, t), $x \in X$, $t = 0 \ldots T\text{-}Max$, augmented with 2 nodes s (*source*) and p (*sink*);
- $E_{T\text{-}Max}$ includes:
  - *Input* arcs $(s, (x, 0))$ and $((x, T\text{-}Max), p)$, with null *vehicle* and *carrier* costs;
  - *idle* arcs $((x, t), (x, t + 1))^{Out}$, with null *vehicle* and *carrier* costs;
  - *carrier-idle* arcs $((x, t), (x, t + 1))^{In}$ with unit *vehicle* costs and *carrier* costs equal to $\beta.Idle\text{-}Cost$ if $x \neq Depot$ and 0 else;
  - *active* arcs $((x, t), (y, t+DIST_{x,y}))$, with *vehicle* costs equal to $\delta.DIST_{x,y}$ and *carrier* cost equal to $\beta.COST_{x,y}$;
  - *backward* arc (p, s) with null *vehicle* costs and *carrier* costs equal to α.

Then, we may set on this network the following multi-commodity flow model:

Network-Flow-VSR Model: {*Compute non negative integral flow vectors F and f, respectively carrier and vehicle flow vectors, such that:*

- *For any idle arc* $e = ((x, t), (x, t + 1))^{Out}$, $f_e \leq C(x)$ *and* $F_e = 0$; (E7)
- *For any carrier-idle arc* $e = ((x, t), (x, t + 1)^{In}$, $f_e \leq CAP.F_e$; (E8)

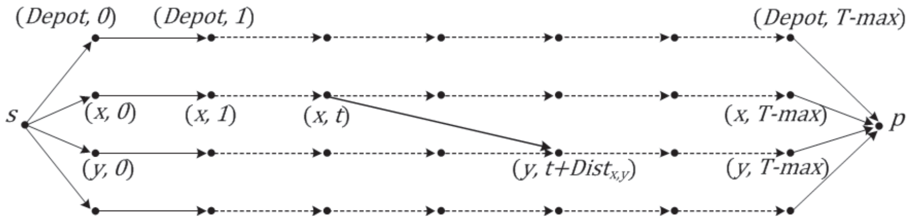

Figure 2 – Dynamic Network $G_{T\text{-}Max} = (X_{T\text{-}Max}, E_{T\text{-}Max})$.

- *For any active arc* $e = ((x, t), (y, t + DIST_{x,y}), f_e \leq CAP.F_e$; (E9)
- *For any* $x \neq Depot$, $F_{(s,(x,0))} = F_{((x,T\text{-}Max),p)} = 0$; (E10)
- *For any node* $x \neq Depot$, $f_{(s,(x,0))} = Sup(v(x), 0)$ *and* $f_{((x,T\text{-}Max),p)} = Sup(-v(x), 0)$; (E11)
- F *and* f *minimize* $Cost_{T\text{-}Max}(F, f) = \Sigma_{e \in ET\text{-}Max} F_e.Carrier\text{-}Cost_e + \Sigma_{e \in ET\text{-}Max} f_e.Vehicle\text{-}Cost_e.\}$

Explanation: *Flow Conservation law* expresses the circulation of *carriers* and *vehicles* between the stations. *Carrier-idle* and *idle* arcs make the difference between *vehicles* which are waiting at some station x while being located either inside some *carrier* or outside. (E9) says that any *vehicle* moving between 2 stations x, y must be contained into some *carrier*. (E10, E11) provide us with initial and final states of both *carriers* and *vehicles*.

Theorem 1. *Solving Network-Flow-VSR is equivalent to solving Preemptive VSR.*

Proof. We first notice that, if a *feasible preemptive VSR tour* collection $\Gamma^* = (\Gamma(k), k = 1..K \leq K\text{-}Max)$ is given, then the *Extended Cost Hypothesis* implies that inequalities (E1) may be supposed to be tight in case $x_i \neq x_{i+1}$. It comes that Γ may be turned into a feasible *Network-Flow-VSR* solution F, f, with same cost, by setting:

- $F_{(p,s)} = K$;
- For any *carrier-idle* arc $e = ((x, t), (x, t + 1))^{In}$:
  - F_e = number of *carriers* k located in x between t and $t + 1$ according to the tours $\Gamma(k)$;
  - f_e = the sum of all quantities $L^*(k)_j$, taken for all *carriers* k as above and j such that $x(k)_j = x$, $x(k)_{j+1} = x$, $T(k)_j \leq t$, $T(k)_{j+1} > t$;
- For any *idle* arc $e = ((x, t), (x, t + 1))^{Out}$: $F_e = 0$ and $f_e = H(\Gamma, x, t)$;
- For any *active* arc $e = ((x, t), (y, t + DIST_{x,y}), x \neq y$:

  - F_e = number of *carriers* k such that $\Gamma(k)$ involves a move from x to y at time t;
  - f_e = sum of all $L^*(k)_j$, for all k as above and j such that $x(k)_j = x$, $x(k)_{j+1} = y$, $T(k)_j = t$;

- For any arc $e = (s, (x, 0))(e = (Depot, T\text{-}Max), p)$, F_e and f_e are defined according to (E10) and (E11).

Conversely, if (F, f) is some *Network-Flow-VSR* feasible solution, then we know that F may be decomposed as a sum of $\{0, 1\}$-valued flow vectors $F(k), k = 1 \ldots K = F_{(p,s)}$. Those elementary flow vectors $F(k), k = 1 \ldots K$, define in a canonical way routes $\Gamma(k)_{Route}$, together with date sequences $\Gamma(k)_{Time}$. Then, for any arc $e = ((x, t), (y, t + DIST_{x,y})$ and $e = ((x, t), (x, t+1))^{In}$, we decompose f_e as a sum of non negative values $f_e(k)$, with values no more than *CAP*. This allows us to deduce *loading sequences* $\Gamma(k)_{Load} = \{L(k)_0, L_1, \ldots, L(k)_{n(\Gamma k))}\}$, $k = 1 \ldots K$, according to a basic $j = 0, \ldots, n(\Gamma(k)$ indexed iterative process. We easily check that the resulting *tour* collection $\Gamma(k), k = 1 \ldots K$ is *preemptive VSR* feasible, with a global cost *Global-Cost*(Γ^*) exactly equal to $Cost_{T\text{-}Max}(F, f)$. □

Let us now try to extend Lemma 0 to this framework. In order to do it, we consider a *carrier* flow vector F and denote by $X(F)$ the node subset of $X_{T\text{-}Max}$ which contains s, p, all nodes $(x, 0)$ and $(x, T\text{-}Max)$, together with all nodes (x, t) which are origin or extremity of some arc $e = ((x, t), (y, t + DIST_{x,y}), x \neq y$, such that $F_e \neq 0$. We provide $X(F)$ with an arc set $E(F)$ which contains arcs $(s, (x, 0))$, $(x, T\text{-}Max), p)$, $x \in X$, as well as:

- related *active* arcs $e = ((x, t), (y, t + DIST_{x,y}), x \neq y$;
- *extended idle* arcs $((x, t), (x, t'))^{In}$ and $((x, t), (x, t'))^{Out}$, with t, t' such that no (x, t'') exists in $X(F)$ such that $t < t' < t''$; those arcs are provided with *vehicle-cost* values respectively equal to $(t' - t)$ and 0;

Since values F_e defined on *idle* arcs $e = ((x, t), (x, t + 1))^{In}$ can be turned in a natural way into values F_e defined on *extended idle* arcs $((x, t), (x, t'))^{In}$, F may be viewed as a flow vector on the network $(X(F), E(F))$. Then we set:

Load-P-VSR: {*Compute, on the network $(X(F), E(F))$ a non negative flow vector f, such that:*

- *For any active arc $e = ((x, t), (y, t + DIST_{(x,y)}), x \neq y$ and any extended-idle arc*

$$e = ((x, t), (x, t'))^{In} \text{ we have } f_e \leq CAP.F_e;$$

- *For any station x, $f_{(s,(x,0))} = Sup(v(x), 0)$ and $f_{((x,T\text{-}Max),p)} = Sup(-v(x), 0)$;*
- *f minimizes the linear cost $\Gamma_{e \in E} f_e.Vehicle\text{-}Cost_e$.*}

This allows us to state the following extension of Lemma 0 (proof left to the reader):

Lemma 1. *F being given, solving Load-P-VSR provides us with an optimal loading strategy.*

One may now ask about casting *Non Preemptive VSR* into this *Network Flow* framework. This leads us to set:

NP-Network-Flow-VSR Model: {*Compute K and two non negative integral multi-commodity flow integral vectors $F = (F(k), k = 1 \dots K)$ and $f = (f(k), k = 1 \dots K)$, respectively carrier and vehicle multi-commodity flow vectors, such that:*

- *For any k, $F(k)$ is $\{0, 1\}$-valued;* (E12)
- *For any idle arc $e = ((x, t), (x, t + 1))^{In}$,*
  $x \neq Depot$, $\Sigma_k f(k)_e = \Sigma_k F(k)_e = 0$; (E13)
- *For any idle arc $e = ((x, t), (x, t + 1))^{Out}$,*
  $x \neq Depot$, $\Sigma_k F(k)_e = 0$; (E13-1)
- *For any idle arc $e = ((Depot, t), (Depot, t + 1))^{Out}$, $\Sigma_k f(k)_e = 0$;* (E13-2)
- *For any active arc $e = ((x, t), (y, t + DIST_{x,y})$, $f(k)_e \leq CAP.F(k)_e$;* (E14)
- *For any $x \neq Depot$, $\Sigma_k F(k)_{(s,(x,0))} = \Sigma_k F(k)_{((x,T\text{-}Max),p)} = 0$;* (E15)
- *For any $x \neq Depot$, $\Sigma_k f(k)_{(s,(x,0))} = Sup(v(x), 0)$ and*
  $\Sigma_k f(k)_{((x,T\text{-}Max),p)} = Sup(-v(x), 0)$; (E16)
- *For any k and any excess (deficit, neutral) station x, values*

  $$f(k)_e, e = ((x, t), (x, t + 1)^{In},$$

  are decreasing (increasing, stationary) when t increases; (E17)
- *F and f minimize $\Sigma_{e \in ET\text{-}Max} F^*_e.Carrier\text{-}Cost_e + \Sigma_{e \in ET\text{-}Max} f^*_e.Vehicle\text{-}Cost_e$.*}

Theorem 2. *Solving NP-Network-Flow-VSR is equivalent to solving Non Preemptive VSR.*

Proof. It is pure routine to check that any *Non Preemptive* VSR feasible solution Γ^* gives rise to a feasible solution (F, f) of *NP-Network-Flow-VSR* with the same cost value. Conversely, monotony constraint (E17) forbids any *carrier k* from unloading (loading) at some *excess* or *neutral* (*deficit* or *neutral*) station x, enabling us to turn flow vector $f(k)$ into a *loading strategy* for the tour $\Gamma(k)$ induced by $\{0, 1\}$-valued flow vector $F(k)$. □

Remark 5. Those reformulations make clearly appear that *Preemption* is a relaxation of *Non Preemption*.

3 VSR LOWER BOUNDS

We propose here 2 classes of easy to compute *vehicle driven* lower bounds for the *VSR* Problem: the first one relies on *Min-Cost Assignment* models which separately bound the *active carrier number*, the *carrier riding cost* and the *vehicle riding time*. The second one directly derives from the previous *Network-Flow-VSR* model.

3.1 Min-Cost Assignment Based Lower Bounds

Let us consider the following ILP models:

VMCA Vehicle-Min-Cost-Assign: {*Compute integral vector* $Q = (Q_{x,y},\ x\ excess,\ y\ deficit) \geq 0$, *such that:*

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

LB-VMCA denotes the related optimal value, which may be computed while relaxing the integrality constraint on the vector Q. In any case (*preemption* or not), *LB-VMCA* provides us with a lower bound of the *vehicle riding time*: $\Gamma_k\ \Sigma_j\ (DIST_{x(k)j,x(k)j+1}.L^*_j)$.

CMCA Carrier-Min-Cost-Assign: {*Compute integral vector* $R = (R_{x,y}, x, y\ stations) \geq 0$, *such that:*

- *For any neutral station* $x \neq Depot$, $\Sigma_y\ R_{x,y} = 0 = \Sigma_y R_{y,x}$
- *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,* $CAP.\Sigma_x\ R_{x,y} = CAP.\Sigma_x\ R_{y,x} \geq -v(y)$
- $\Sigma_y\ R_{Depot,y} = \Sigma_y\ R_{y,Depot} \geq 1$
- *Minimize* $\Sigma_{x,y}\ COST_{x,y}.R_{x,y}\}$

LB-CMCA denotes the related optimal value, which may be computed in polynomial time through a simple *Min Cost Flow* algorithm. *LB-CMCA* is a lower bound for the *carrier riding cost* Σ_k*L-E-COST*$(\Gamma(k))$. If *LB-Time-CMCA* is the value of the *CMCA* model obtained by replacing the *COST* matrix by the *DIST* matrix, then *LB-Time-CMCA/T-Max* is a lower bound for the *carrier number* K.

UCMCA Unit-Carrier-Min-Cost-Assign: {*Compute rational vector* $R = (R_{x,y},\ x stations) \geq 0$, *such that:*

- *For any neutral station* $x \neq Depot$, $\Sigma_y R_{x,y} = 0 = \Sigma_y R_{y,x}$
- *For any* x, y, *both deficit or both excess,* $R_{x,y} = 0$
- *For any* y *deficit,* $R_{Depot,y} = 0$ *and for any* x *excess,* $R_{x,Depot} = 0$
- *For any excess station* x, $\Sigma_{y \text{ deficit or Depot}} R_{y,x} = \Sigma_{y \text{ deficit}} R_{x,y} = v(x)$
- *For any deficit station* y, $\Sigma_{x \text{ excess or Depot}} R_{y,x} = \Sigma_{x \text{ excess}} R_{x,y} = -v(y)$
- $\Sigma_{y \text{ excess}} R_{Depot,y} = \Sigma_y R_{y \text{ deficit},Depot} = 1$
- *For any subset* $A \subseteq X - \{Depot\}$, $A \neq Nil$, $\Sigma_{x \in Ay, y \notin A} R_{x,y} \geq 1$ *(No-Subtour Constraint)*
- *Minimize* $\Sigma_{x,y}$ $COST_{x,y}.R_{x,y}\}$

LB-UCMCA denotes the related optimal value. We see that *LB-UCCA* is a lower bound for the *L-COST* value of any tour γ which starts and ends into *Depot*, while alternatively moving from *excess* nodes to *deficit* nodes and carrying unit loads. If *LB-Time-UCMCA* is the optimal value of the *UCMCA* model obtained by replacing the *COST* matrix by the *DIST* matrix, then *LB-Time-UCMCA* is a lower bound for the *carrier riding time L-DIST* induced by γ. *UCMCA* involves significantly less variables than the *CMCA* model.

We deduce:

Theorem 3. *A VSR (Preemptive or Not) lower bound is given by LB-MCA* $= \alpha$ *LB-Time-CMCA/T-Max* $+ \beta$*.LB-CMCA* $+ \delta$*.LB-VMCA.*

Proof. It is contained into the comments which come together with the definition of the above models. □

Theorem 4. *A Non Preemptive VSR lower bound is given by LB-UMCA* $= \alpha$ *LB-Time-UCMCA/ (CAP.T-Max)* $+ \beta$*.LB-UCMCA/CAP* $+ \delta$*.LB-VMCA.*

Proof. Any *tour* γ which satisfies (E1, E2, E3, E4) may be split into *CAP tours* $\gamma_1, \ldots, \gamma_{CAP}$, all with same lengths, which globally perform the *relocation* process when related $CAP = 1$. It comes from the fact that any solution Z of the *LOAD-NP-VSR* model related to γ may be decomposed into a sum $Z_1 + \cdots + Z_{CAP}$ of {0, 1}-valued flow vectors. 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 $T\text{-}Max = +\infty$, we see that: the *carrier riding time (carrier riding cost)* of any solution Γ of *Non Preemptive VSR* is at least equal to *Carrier-Ride-Time*$_1$*/CAP (Carrier-Ride-Cost*$_1$*/CAP)*. We deduce that $\alpha.\lceil$*Carrier-Ride-Time*$_1$*/CAP.T-Max*$\rceil + \beta$ *Carrier-Ride-Cost*$_1$*/CAP* + δ*.LB-VMCA* is a *Non Preemptive VSR* lower bound. But *Carrier-Ride-Time*$_1$ corresponds to a kind of TSP *carrier* tour starting and ending into *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*$_1$. So we conclude. □

3.2 Projected Flow Lower Bound

We derive from the dynamic network $G_{T\text{-}Max} = (X_{T\text{-}Max}, E_{T\text{-}Max})$ of Section 2.4 a *projected* network $G_{Proj} = (X_{Proj}, E_{Proj})$ as follows (see Fig. 3):

- $X_{Proj} = X \cup \{s, p\}$ where nodes s and p are additional nodes *source* and *sink*;
- The restriction of G_{Proj} to X is a complete network: any arc $e = (x, y)$ is provided with a *carrier* cost $CC_e = \beta.COST_{x,y} + (\alpha/T\text{-}Max).DIST_{x,y}$ and with 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.

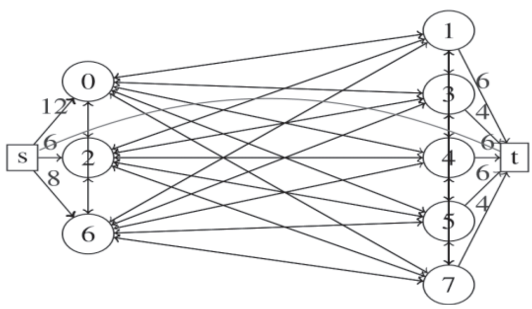

Figure 3 – A network G_{Proj} derived from 3 excess stations and 5 deficit stations.

Then we set:

Projected-VSR-Flow Model: {*Compute on the network G_{Proj} two integral flow vectors H and h such that:*

- *For any arc* $e = ((x, y), x, y \neq s, p, h_e \leq CAP.H_e$ (E18)
- *For any excess (or neutral) station* x, $h_{(s,x)} = v(x)$ *and for any deficit station* y, $h_{(yp)} = -v(x)$ (E19)
- $\Sigma_y H_{Depot,y} = \Sigma_y H_{y,Depot} \geq 1$ (E20)
- *Minimize* $\Sigma_e CC_e.H_e + \Sigma_e CV_e.h_e$.}

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

Theorem 5. *LB-Proj-Flow is a (Preemptive or Not) VSR lower bound, such that LB-Proj-Flow $\geq$ LB-MCA.*

Proof. Any feasible solution F, f of *Network-Flow-VSR* can be turned into a feasible solution H, h of *Projected-VSR-Flow*. The cost $Cost_{T\text{-}Max}(F, f) = \Sigma_{e \in ET\text{-}Max} F_e.Carrier\text{-}Cost_e + \Sigma_{e \in ET\text{-}Max}\ f_e.Vehicle\text{-}Cost_e$ may be decomposed into:

$$Cost_{T\text{-}Max}(F, f) = \alpha.F_{(p,s)} + \Sigma_{e \in ET\text{-}Max, e \neq (p,s)} F_e.Carrier\text{-}Cost_e + \Sigma_{e \in ET\text{-}Max}\ f_e.Vehicle\text{-}Cost_e.$$

Through projection, the two last terms of this sum give rise to the quantity $\Sigma_{arcs\ e} \beta.COST_e.H_e + \Sigma_{arcs\ e} CV_e.h_e$. The first component corresponds to $\alpha.K$, where $K = F_{(p,s)}$ is the *carrier number*. But we know that this *carrier number* is at least equal to $(\Sigma_{e=(x,y) \in E\text{-}Proj}\ H_{x,y}.DIST_{x,y})/T\text{-}Max$. We deduce the first part of our statement.

As for the second part, we get it by noticing that any solution H, h of the *Projected-VSR-Flow* program give rise in a natural way to a feasible solution R of the *CMCA* program and a feasible solution Q of the *VMCA* program, and by keeping on with the above decomposition of the quantity $\Sigma_{arcs\ e} CC_e.H_e + \Sigma_{arcs\ e} CV_e.h_e$. □

Remark 6. The *Projected-VSR-Flow* model does not solve our *VSR* problem, even according to its *preemptive* version. For instance one may consider a *station* set $X = \{Depot, A, B, C\}$, a *carrier* flow H related to the *route* $(Depot, A, B, C, A, Depot)$ followed by 1 *carrier* with capacity 1, and a *vehicle* flow h 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 7. *LB-Proj-Flow* value provides us with a better lower bound than the *LB-MCA* lower bound of Theorem 3. Still, *Projected-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 (E18) yields a *Lagrangean* value $\text{Sup}_{\lambda \in \Lambda}(\text{Inf}_h(CV + \lambda).h) + \text{Inf}_H(CC - \lambda).H)$ where:

- Vector flow h is subject to (E19) and vector flow H is subject to (E20);
- $\Lambda = \{\lambda$ such that the restriction of the graph G_{Proj} to X does not contain any negative $(CC - \lambda)$-circuit$\}$.

But, because of the total unimodularity of flow constraint matrices, this value is the same as the value obtained by performing *Lagrangean* relaxation of (E18) on the rational relaxation of *Projected-VSR-Flow*. That means that the above Lagrangean value does not improve the standard relaxation of the integrality constraint.

4 A VEHICLE MIN-COST ASSIGNMENT BASED HEURISTIC FOR NON PREEMPTIVE VSR

We focus here on *Non Preemptive VSR*, and derive from the *LB-MCA* lower bound a decomposition of this problem into a Master *Vehicle-Min-Cost Assignment* problem and a Slave *Pick-up&Delivery (PDP)* Problem.

4.1 MCA/PDP Decomposition

Let us recall that a *Pick-up&Delivery* instance (see [2, 12, 17]) 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; N denotes the set of all nodes $o(j)$, $d(j)$, $j \in J$, augmented with a *Depot* node and considered as pairwise distinct; these *requests* have to be served by *trucks*, initially located in *Depot* and all with capacity *CH*;
- 2 distance matrices *D* and *CS*, indexed on the set $N.N$ and a threshold *D-Max*;
- Scaling coefficients *A*, *B*, *C*.

A collection ρ of *truck routes* $\rho(m)$, $m = 1 \ldots M$ defined on the set N is a *feasible PDP* solution if:

- every *request* j is serviced by some *truck* m : m first loads $\lambda(j)$ at $o(j)$ and unloads it into $d(j)$;
- the load of a *truck* never exceeds capacity *CH*;
- the *D*-length *of* $\rho(m)$ of any *truck route* $\lambda(m)$, $m = 1..M$, never exceeds *D-Max*.

It is an optimal *PDP* solution if it is *feasible* and minimizes a quantity:

$$\textit{PDP-COST}(\rho) = A.M + B.\Sigma_m \textit{CS-Length}(\rho(m)) + C.\Sigma_j \lambda(j).\textit{D-Ride}(j),$$

where *D-Ride*(j) is the *D*-length which is run by load $\lambda(j)$ inside a *truck*. A *Load-Split PDP* instance is defined the same way, but every loads $\lambda(j)$ may be split into a sum $\lambda(j) = \lambda(j)_1 + \cdots + \lambda(j)_{Q(j)}$, of several sub-loads, which are separately handled.

Though *Load-Split PDP* is NP-Hard, it may be in practice efficiently handled through a GRASP-VNS (*Greedy Randomized Adaptative Search + Variable Neighborhood Search*) process based upon *Insert/Remove* operators:

- *Insert* operator: Inserting *request* $j = (o(j), d(j), \lambda(j))$ into some *truck 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 and the $\lambda(j)$ value accordingly.

Then related GRASP-VNS scheme comes as follows:

PDP GRASP-VNS Algorithm

Randomized Initialization:

While all *requests* have not been inserted do

Randomly pick up some non inserted *request* j;

Compute (in a heuristic way) *truck* parameter m, together with insertion parameters $x, y \in \rho(m)$, and $\lambda \leq \lambda(j)$ in such a way that related insertion is feasible and such that (bi-criteria choice):

- the induced increase of *PDP-COST*(ρ) is the smallest possible;
- λ is the largest possible;

Local Search Loop:

Not *Stop*;

While Not *Stop* do

Identify a set $J_0 \subseteq J$ of *poorly inserted requests*;

Remove J_0 from J and reinsert it according to the same process as in the initialization;

Update the current best solution $\rho^* = (\rho(m), m = 1..M)$; Update *Stop*.

Let us now come back to our *Non Preemptive VSR* instance, and suppose that, for some instance (*X*, *v*, *C*, *CAP*, *T-Max*, *DIST, COST*), we know, for every pair (x, y), x *excess*, y *deficit station*, which quantity $Q_{x,y}$ has to move from x to y. Then, we only need to solve the *Load-Split PDP* instance defined by:

- *Requests* j *are all 3-uples* $(o(j) = x, d(j) = y, \lambda(j) = Q_{x,y})$*, taken for all pairs* x, y *such that* $Q_{x,y} \neq 0$;
- *D-Max* = *T-Max*; *D* = *DIST*; *CS* = *COST*; *CH* = *CAP*; $A = \alpha, B = \beta$, $\mathrm{C} = \delta$.

One may conjecture 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 *VMCA*(U) (*Vehicle Min-Cost Assignment*) model:

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

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

Though we cannot prove this conjecture, it leads us to the following reformulation of *Non Preemptive VSR*:

Non Preemptive VSR VMCA Reformulation: {*Compute cost vector* $U = (U_{x,y}, x\ Excess,\ y\ Deficit) \geq 0$, such that the optimal value of the related Load-Split PDP instance be the smallest possible}.

We may handle this reformulation through the following algorithmic scheme:

VSR-MCA Algorithm *(N: Loop Number)*

Initialize cost vector $U = (U_{x,y}, x\ Excess,\ y\ Deficit) \geq 0$;

For $j = 1 \ldots N$ do (**Local Search* loop*)

Derive a *PDP Assignment* vector Q through optimal resolution of *VMCA*(U);

Solve (in a heuristic way) the related *Load-Split PDP* instance;

Update cost vector U;

Apply to the resulting *route* collection $\Gamma^*_{Route} = \{\Gamma_{Route}(1), \ldots, \Gamma_{Route}(K)\}$ the *Load-NP-VSR* model, clean the routes $\Gamma_{Route}(k)$ from its useless stations;

Keep the best result ever obtained.

Two critical points have to be specified inside this algorithmic description:

1) "Initialize cost vector *U*" instruction: LB-MCA lower bound of Section 3 suggests us to apply what we call the *Shortest Distance/Cost Strategy*, and set, 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 non negative coefficient; as a matter of fact, doing this leads us to extend the above VSR-MCA algorithm into a GRASP algorithmic scheme, by performing initialization of the cost vector U in a random way:

GRASP-VSR-MCA Algorithm *(N: Loop Number, R: Replication Number)*

For $i = 1 \ldots R$ do

Randomly generate $\lambda \geq 0$;

For any x, y, x *Excess*, y *Deficit*, $U_{x,y} \leftarrow DIST_{x,y.} + \lambda.(COST_{x,y.} + COST_{y,x})$;

For $j = 1 \ldots N$ do ... (**Local Search* loop of *VSR-MCA**);

Keep the best result ever obtained.

2) "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 *Non Preemptive VSR* solution Γ^*, whose global cost *Global-Cost*(Γ^*) 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-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-Cost*$(\Gamma *)$ may be written *Global-Cost*$(\Gamma^*) = \Sigma_{r \in Req(U)}$ *Partial-Cost* (r, Γ^*), where *Partial-Cost*(r, Γ^*) is the part of *Global-Cost*(Γ^*) which is charged this way to request r. Then, for every request $r = (x, y, Q_{x,y} \neq 0)$ we set $V_{x,y} = \textit{Partial-Cost}(r, \Gamma^*)/Q_{x,y}$ 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 $= \tau.U^0_{x,y}$.

4.2 An Approximation Result

A natural question comes about the quality of the *Shortest Cost/Distance strategy*. Since, in most cases, the *COST* and *DIST* matrices are strongly correlated, we consider here the case when those matrices are the same, and when *Global-Cost* only involves the *carrier riding cost*. In such a case, we may state:

Theorem 6 (Shortest Cost/Distance Strategy). *If COST = DIST, if* $\alpha = \delta = 0$ *(focus on carrier riding cost minimization) and if T-Max* $= +\infty$*, then the Shortest Cost/Distance strategy induces an approximation ratio of (1+CAP). This is the best possible ratio.*

Proof. We first notice that we may, since *T-Max* $= +\infty$, deal with only one *carrier*. Let us first prove the first part of the result, that means that there is no approximation ratio better than (1+*CAP*). In order to do so, we build the following *Non Preemptive VSR* instance:

– $K = 1$;

– $X = \{Depot\} \cup \{o_{n,c}, d_{n,c}, n = 0..N\text{-}1, c = 1..CAP\}$ where N is a large number; function v is equal to 1 for $o_{n,c}$ *(excess) stations* and to -1 for $d_{n,c}$ (*deficit*) stations;

- $DIST = COST$ represents the shortest path distance induced on the set X by the following arc set $E = E_1 \cup E_2 \cup E_3 \cup E_4$:
  - $E_1 = \{(Depot, o_{0,1}), (d_{N\text{-}1,1}, Depot)\}$, both arcs with length equal to $1/2$;
  - $E_2 = \{(o_{n,c}, o_{n,c+1}), (d_{n,c+1}, d_{n,c}), n = 0..N\text{-}1, c = 1..CAP-1\}$, all arcs with *small* length ε;
  - $E_3 = \{(o_{n,CAP}, d_{n,CAP}), n = 0..N\text{-}1\} \cup \{(d_{n,1}, o_{n+1,1}), n = 0..N\text{-}2\}$, all arcs with length 1;
  - $E_4 = \{\{(o_{n,c}, d_{n-1,c})\}, n = 0..N\text{-}1, c = 1..CAP\}$ addition being performed modulo N, all arcs with length 1-α, where α is a small number.

One easily checks that an optimal *tour* for the *carrier* is the *tour* $\{Depot, o_{0,1}, \ldots, o_{0,CAP}, d_{0,CAP}, \ldots, d_{0,1}, o_{1,1,\ldots}, o_{1,CAP}, \ldots, Depot\}$, with length $L\text{-}DIST = 2n + 2n.(CAP\text{-}1)\varepsilon$. For every $n = 0, \ldots, N\text{-}1$, this *tour* makes the *carrier* load all the *excess vehicles* located in *excess* stations $o_{n,c}$, $c = 1 \ldots CAP$, and next bring them to *deficit* stations $d_{n,c}$, $c = CAP \ldots 1$, before moving to node $o_{n+1,1}$. On another side, the vector Q deriving from the *Shortest Cost/Distance* strategy is provided by E_4. One checks that a related optimal *PDP* meets every request related to an arc $(o_{n,c}, d_{n-1,c})$ through a direct move $(o_{n,c}, d_{n-1,c})$ (proof left to the reader: if it were not the case, then one could remove related arcs of E_4). So, as soon as the carrier has been loading in station $o_{n,c}$, it moves to station $d_{n-1,c}$ and delivers its load. A consequence is that at any time during the process, the current loads of the *carrier* does not exceeds 1 and that the optimal *PDP* solution comes as a sequence $\{Depot, o_{0,1}, d_{n-1,1}, o_{0,2}, d_{n-1,1}, \ldots, d_{n-1,CAP}, o_{1,1,\ldots} d_{0,1}, \ldots, Depot\}$, with length $L\text{-}DIST = CAP.n(1-\alpha) + n.CAP.(1 + (CAP\text{-}1).\varepsilon + 2n + n.(CAP\text{-}1)\varepsilon$. We conclude.

In order to prove the first part of the result, that means that $(1 + CAP)$ provides us with an approximation ratio, we first notice that splitting any station x into $v(x)$ copies, all with v value equal to 1 or -1 and to distance 0 to each other does not modify the problem. Then we consider some feasible *Preemptive VSR* tour $\gamma = \{Depot, x_0, x_1, \ldots, x_{n(\gamma)} = Depot\}$. Clearly, we may suppose that no station is involved more than once in γ. Then we may state:

Lemma 2. *There cannot exist any sequence (*discrete circular interval*) $J = \{x_i, x_{i+1} \ldots, x_{i+t}\}$, addition being taken modulo $n(\gamma)$, such that $\Sigma_{x \in J} v(x) \leq CAP - 1$.*

Proof. If such a sequence exists then the load of the *carrier* just before reaching x_i is at least equal to $CAP+1$. □

Lemma 3. *There exists some one-to-one involutive correspondence $u = u_\gamma$ from X into itself such that:*

- *If x is an excess station then $u_\gamma(x)$ is a deficit station and conversely;*
- *If one runs along γ from some deficit station x, then it visits no more that $CAP-1$ stations other than (eventually) Depot, x and $u_\gamma(x)$ before reaching $u_\gamma(x)$. We denote by $\gamma(x, u)$ the related sub-path of γ.*

By the same way there exists a one-to-one involutive correspondence $w = w_\gamma$ from X into itself such that:

- *If x is an excess station then $w_\gamma(x)$ is a deficit station and conversely;*
- *If one runs along γ from some excess station x, then it visits no more that CAP-1 stations other than (eventually) Depot, x and $w_\gamma(x)$ before reaching $w_\gamma(x)$. We denote by $\gamma(x, w)$ the related sub-path of γ.*

Proof. For any node $x = x_i$ of γ, we set $J_x = \{x_i, x_{i+1}, \ldots, x_{i+CAP}\}$, addition being taken modulo $n(\gamma)$. Then, we build a bipartite graph (U, V, E) by setting:

- $U = \{$*deficit* stations of $\gamma\}$; $V = \{$*excess* stations of $\gamma\}$;
- $E = \{(x_i, x_j)$ such that one visits no more than *CAP*-1 non trivial stations when running from x_i to x_j along $\gamma\}$.

The first part of Lemma 3 (existence of $u = w_\gamma$) means that this bipartite graph admits a perfect matching. If it is not true, then Koenig-Hall Theorem tells us that there exists $U^* \subseteq U$ such that Card($\{v \in V$ which are the extremity of an edge $(u, v), u \in U^*\}) \leq$ Card$(U^*) - 1$. One may choose U^* in such a way that the intersection graph defined by the discrete circular intervals $J_x, x \in U^*$ is connected. But then we see that the discrete interval $J = \cup_{x \in U^*} J_x$ is such that $\Sigma_{x \in J} v(x) \leftarrow CAP$, and thus that it contradicts former Lemma 2. We proceed the same way in order to get the existence of $w = w_\gamma$. □

Lemma 4. *A same transition $x_i \rightarrow x_{i+1}$ ($i + 1$ being computed modulo n) of $\gamma = \{Depot, x_0, x_1, \ldots, x_n = Depot\}$, cannot appear more than CAP times in the path collection $\{\gamma(x, u), \gamma(x, w), j = 0, \ldots, n - 1\}$ of Lemma 3.*

Proof. If the transition $x_i \rightarrow x_{i+1}$ is involved into $\gamma(x, u)$ then x is a *deficit* station and is one of the *CAP* stations which are located before in γ. If it is involved into $\gamma(x, w)$ then x is an *excess* station and is one of the *CAP* stations which are located before in γ. We conclude. □

We may now finish with the proof of Theorem 6. Let us suppose that *tour* γ is an optimal solution of *Non Preemptive VSR* and that we are provided with a *min-cost assignment* Q, which, with any *excess* station x, associates some *deficit* station $z_Q(x)$ in a one-to-one way and which is such that $\Sigma_{x\ excess} DIST_{x, Q(x)}$ is the smallest possible. Then for any *excess* station x, we may derive a circuit $\gamma(x)$ as follows: Start from x, then go to the *deficit* node $z_Q(x)$, next go to $u_\gamma(x)$ of Lemma 3 while following γ and keep on this way. Two circuits $\gamma(x)$ and $\gamma(y)$ are either identical or disjoint, and induce a partition of $X - \{Depot\}$ into a collection $\{\gamma_1, \ldots, \gamma_P\}$ of circuits, with related *representative* stations x^p labeled in such a way that they come according to this order in the tour γ.

Then we derive a new *Non Preemptive VSR* solution γ^* as follows: Start from *Depot*, go to γ_1 representative station x^1 along γ, run along γ_1, next go to x^2 along γ and so on until going back to *Depot* after running γ_P.

The length $L\text{-}DIST(\gamma^*)$ is equal to $L\text{-}DIST(\gamma) + \Sigma_p\ L\text{-}DIST(\gamma_p)$. But we also have $\Sigma_p\ L\text{-}DIST(\Sigma_p) = \Sigma_{x\ deficit}\ L\text{-}DIST(\gamma(x,u)) + \Sigma_{x\ excess}\ DIST_{x,zQ(x)} \leq \Sigma_{x\ deficit}\ L\text{-}DIST(\gamma(x,u)) + \Sigma_{x\ excess}\ L\text{-}DIST(\gamma(x,w))$. Because of Lemma 4, a same transition $x_i \rightarrow x_{i+1}$ of $\gamma = \{x_0 = Depot, x_1, .., x_{n(\gamma)} = Depot\}$, cannot appear more than *CAP* times. We deduce $\Sigma_{x\ deficit}\ L\text{-}DIST(\gamma(x,u)) + \Sigma_{x\ excess}\ L\text{-}DIST(\gamma(x,w)) \leq CAP.L\text{-}DIST(\gamma)$ and we conclude. □

5 A PROJECTED FLOW BASED HEURISTIC FOR NON PREEMPTIVE VSR

We still focus here on *Non Preemptive VSR* problem, and derive from the *LB-Proj-Flow* lower bound a heuristic scheme which relies on the reconstruction, from a *Projected-VSR-Flow* solution, of a feasible *Non Preemptive VSR* solution. In order to describe it, we first introduce a *feasibility oriented* version of the *Load-NP-VSR* model:

Feasibility-Load-NP-VSR Model: {*Given a route collection* Γ^*_{Route}*, compute on the network* $H(\Gamma^*_{Route})$ *of Section 2.2 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*, $Z_e \leq v(x)$; *for any arc* $e = (Def(y), p)$, *y deficit,* $Z_e \leq -v(x)$;
- *Maximize* $Z_{p,s}$}.

Then a synthetic description of our heuristic scheme comes as follows:

Projected-Vehicle-Flow Algorithm

$\Gamma^*_{Route} \leftarrow Nil$;

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

Compute an optimal solution (H, h) of the *Projected-VSR-Flow* model; (I1)

Derive a *route* collection $\gamma^*_{Route} = \{\gamma_{Route}(1), \ldots, \gamma_{Route}(P)\}$ from H; (I2)

Apply *Feasibility-Load-NP-VSR* to the *route* collection $\gamma^*_{Route} \cup \gamma^*_{Route}$ and get a resulting flow vector Z; $\Gamma^*_{Route} \leftarrow \Gamma^*_{Route} \cup \gamma^*_{Route}$. Accordingly update coefficients $v(x), x \in X : v(x) \leftarrow v(x) - Z_{(s,Exc(x))}$;

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

We must now go further into the description of key instructions (I1) and (I2):

– (I1): *Handling of the VSR-Flow model*: We do it here through the use of a MIP library, while imposing a threshold on the computing time, as soon as the number of stations exceeds 30.

– (I2): *Derive a route collection* $\gamma^*_{Route} = \{\gamma_{Route}(1), \ldots, \gamma_{Route}(P)\}$ *from* H *and* h: Flow vector H defines a collection of arcs (x, y), each of them taken $H_{(x,y)}$ 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, $j = 1 \ldots s$, of the resulting graph gives rise to some *Eulerian* route γ_j. Then we build γ^*_{Route} by starting from *Depot*, reaching some closest X_j into some node x_j, running γ_j until being back to x_j and keeping on with another connected component X_j. Every time the length *L-DIST* of current route $\gamma_{Route}(p)$ is on the edge to exceed the *T-Max* threshold, we close it and start $\gamma_{Route}(p+1)$.

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:

Route-Reconstruction Algorithm:

Input: the Flow vector H, and the $v(x)$, $x \in X$ coefficients;

Initialization: For any $x, u(x) \leftarrow v(x)$; $P \leftarrow 1$; *H-Cour* $\leftarrow H$; *Penalty* $\leftarrow 0$; *Profit* $\leftarrow 0$;

While *H-Cour* $\neq 0$ do

Not *Stop*; *x-cour* $\leftarrow$ *Depot*; $\gamma_{Route}(P) \leftarrow$ {*Depot, Depot*}; *Load* $\leftarrow 0$; *Length* $\leftarrow 0$;

While Not *Stop* do

<u>1th case</u>: There exists at least one station y such that:

$$Length + DIST_{x\text{-}cour,\ y} + DIST_{y,\ Depot} \leq T\text{-}Max; \qquad (*)$$

$$\text{and} \quad H_{x\text{-}cour,\ y} \neq 0; \qquad (**)$$

For any such a station y, compute $L(y) = \text{Inf}(CAP - Load, v(y))$ in case y is *excess*, and $L(y) = \text{Inf}(Load, -v(y))$ in case y is *deficit*;

Pick up y_0 which satisfies $(*)$ and $(**)$ and is such that $L(y_0)$ is maximal;

Move to y_0 : $u(y_0) \leftarrow u(y_0) - L(y_0)$; *Load* $\leftarrow$ *Load* $+L(y_0)$; $H_{x\text{-}cour,\ y0} \leftarrow H_{x\text{-}cour,y0} - 1$; *Length* $\leftarrow$ *Length* $+ DIST_{x\text{-}cour,\ y0}$; *Profit* $\leftarrow$ *Profit* $+L(y_0)$; *x-cour* $\leftarrow y_0$;

<u>2th case</u>: 1th case does not hold, but there exists y such that $(*)$;

For any such y compute $L(y)$ as above;

Pick up station y_0 which satisfies $(*)$ and is such that (bi-criteria choice):

- $L(y_0)$ is large and $COST_{x,y0}$ is small;

Move to y_0 as *I* the first case with $H_{x\text{-}cour,\ y0}$ unchanged and *Penalty* $\leftarrow$ *Penalty* $+ COST_{x,y0}$;

3th case: None among previous cases 1 and 2 holds;

If x-*cour* is an *excess* station, then *move* back along $\gamma_{Route}(P)$ until x-*cour* is a *deficit* station; Close current route $\gamma_{Route}(P)$ by coming back from x-*cour* to *Depot*; *Stop*;

$P \leftarrow P + 1$;

Remark 8. *Route-Reconstruction* aims at building γ_{Route} in such a way it maximizes *Profit* and minimizes both *Penalty* and *P*. Tree search would be too costly. Instead, we randomize *Route-Reconstruction* and launch it several times, before keeping the best collection γ_{Route} ever obtained.

6 A FLOW RECONSTRUCTION HEURISTIC FOR PREEMPTIVE VSR

We deal now with the *preemptive* version of *VSR*, and involve the *Dynamic Network Framework* of Section 2.4, according to the following algorithmic scheme:

Flow-Reconstruction-P-VSR Algorithm:

1th step: Compute an optimal solution (H, h) of the *VSR-Flow* model;

2th step: Denote by G^h the network induced by non null $h_{x,y}$ values; Because of the optimality of (H, h), G^h does not contain any circuit; Add 2 nodes $Depot_1$ *and* $Depot_2$ to G^h and:

- Connect $Depot_1$ to any minimal node (which admits no predecessor but s) $x \neq s$ of G^h;
- Connect any maximal node (which admits no successor but p) $y \neq p$ of G^h to $Depot_2$;
- Provide related arcs with *DIST* values in a natural way;

Denote by $G*^h$ the resulting network; flow vector h may be considered as defined on $G*^h$;

3th step: Compute largest paths, according to *DIST*, respectively from $Depot_1$ to any node x of $G*^h$, and from any node y of to $Depot_2$; Denote by $L\text{-}DIST_x$ and $L\text{-}DIST*_y$ the resulting *DIST*-length values; In case $x = s$, set $L\text{-}DIST_s = 0$ and do as if any arc (s, x) where provided with null *DIST* value; Do the same thing with p and $L\text{-}DIST*_p$;

4th step: Until $L\text{-}DIST_{Depot2} \leq T - Max$ do *Refine* $G*^h$ and h;

5th step: Derive from h a flow vector f defined on the dynamic network $G_{T\text{-}Max} = (X_{T\text{-}Max}, E_{T\text{-}Max})$ and which satisfies (E7, E11) of the *Network-Flow-VSR* model;

6th step: Derive a feasible solution (F, f) of the *Network-Flow-VSR* model.

Let us now describe into more details the contents of steps 4, 5 and 6.

– **Step 4: *Refine* procedure**.

Let us consider some node $x \neq s$, p, $Depot_1$, $Depot_2$ and such that $L\text{-}DIST_x + L\text{-}DIST^*_x = L\text{-}DIST_{Depot2}$ (*critical* node), together with some integral number w between 1 and $(\Sigma_y h_{x,y}) - 1$. We may rank predecessors (successors) y of x according to increasing $L\text{-}DIST_y + DIST_{y,x}$ values (decreasing $L\text{-}DIST^*_y + DIST_{x,y}$ values). Then we define the *Split* procedure as follows (see Fig. 4):

Procedure *Split*(x, w):

Make two copies x' and x'' of x;

Assign flow values $h_{y,x'}$ to the arcs (y, x'), y predecessor of x, in such a way that:

- they do not exceed $h_{y,x}$ values;
- $\Sigma_y h_{y,x'} = w$;
- the vector $h^{x'} = (h_{y,x'}, y$ predecessor of $x)$ is maximal according to the lexicographic order related to above defined ranking;

Assign remaining flow values $h_{y,x} - h_{y,x'}$, y predecessor of x, to the arcs (y, x'');

Do the same thing with arcs (x', y) and (x'', y), y successor of x, while taking into account that successors of x are ranked through decreasing $L\text{-}DIST^*_y + DIST_{x,y}$ values;

Delete node x; Delete arcs (y, x') and (x'', y) which are provided with null flow values $h_{y,x'}$, $h_{x'',y}$;

Compute:

- $\Delta' = \text{Sup}_{y \text{ predecessor } x'}(L\text{-}DIST_y + DIST_{y,x'}) + \text{Sup}_{y \text{ successor } x'}(L\text{-}DIST^*_y + DIST_{x',y})$;
- $\Delta'' = \text{Sup}_{y \text{ predecessor } x''}(L\text{-}DIST_y + DIST_{y,x''}) + \text{Sup}_{y \text{ successor } x'}(L\text{-}DIST^*_y + DIST_{x'',y})$;
- $\Delta = \text{Sup } \Delta', \Delta''$. (N.B: Δ should be no larger than $L\text{-}DIST_{Depot2}$)

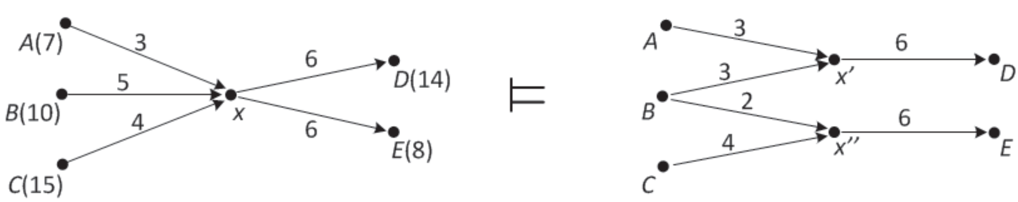

Figure 4 – The Split Mechanism.

Then the *Refine* procedure comes as follows:

Procedure *Refine*:

Compute x and w in such a way that resulting Δ value be the smallest possible;

Replace current network G^{*h} and related flow vector h by the graph and flow vector which derive from application of $Split(x, w)$;

Update $L\text{-}DIST_y$ and $L\text{-}DIST{*}_y$ values;

– **Step 5: Construction of flow vector** f

To every node $x^* \neq Depot_1, Depot_2$ in the *refined* graph G^{*h}, correspond both a station x and a time value $t = L\text{-}DIST^*_x$. That means that we may associate with x^* a node (x, t) of the network $G_{T\text{-}Max}$. In case (x^*, y^*) is an arc of G^{*h} such that related nodes (x, t), (y, u) of $G_{T\text{-}Max}$ satisfy $u - t > DIST_{x,y}$ then we insert a node $(x, u - DIST_{x,y})$, and split the arc (x^*, y^*) into two arcs $((x, t), (x, u - DIST_{x,y}))$, and $((x, u - DIST_{x,y}), (y, u))$. Flow vector h is updated accordingly (see Fig. 5).

Once this has been done, we consider, for any station x, all nodes (x, t) of $G_{T\text{-}Max}$ which have been created this way, add nodes $(x, 0)$ and $(x, T\text{-}Max)$, and rank all those nodes through increasing t values ($t_0 = 0, \ldots, t_I = T\text{-}Max$). Then we replace arcs $((x, t)$, $(x, u - DIST_{x,y,}))$ by arcs $((x, t_i), (x, t_{i+1}))^{In}$ and $((x, t_i), (x, t_{i+1}))^{Out}$, $i = 0, \ldots, t_{I-1}$, and distribute in a natural way all flow values $h_{(x,t),(x,u)}$, $u > t$, previously obtained, between arcs $((x, t_i), (x, t_{i+1}))^{In}$ and $((x, t_i), (x, t_{i+1}))^{Out}$. We do it in a way which is consistent with both capacities CAP and C, and which minimizes the sum of flow values h on the arcs $((x, t_i), (x, t_{i+1}))^{In}$. Finally, we remove nodes $Depot_1$ and $Depot_2$, and assign flow values to arcs $(s, (x, 0))$ and arcs $((y, T\text{-}Max), p)$ in such a way relations (E10, E11) of the *Network-Flow-VSR* model be satisfied. By doing this, we turn h into a flow vector f, which may be considered as defined on an implicit representation of $G_{T\text{-}Max}$ and which satisfies (E7, E11) of the *Network-Flow-VSR* model. We check that the resulting cost $\Sigma_e \in ET\text{-}Max f_e.Vehicle\text{-}Cost_e$ only differs from initial $\Sigma_{arcs\, e} CV_e.h_e$ by the time *vehicles* spend inside the *carriers* on arcs $((x, t_i), (x, t_{i+1}))^{In}$.

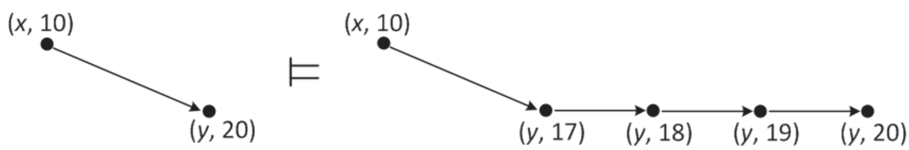

Figure 5 – Decomposing an arc $((x, 10), (y, 20))$ such that $\text{DIST}_{x,y} = 7$.

– **Step 6: Construction of flow vector** F

We complete the construction of step 5 by introducing nodes $(Depot, 0)$, $(Depot, T\text{-}Max)$ and all nodes $(Depot, t)$ such that there exists (x, u), obtained through step 5 and such that $u = t + DIST_{Depot,x}$ or $t = u + DIST_{x,Depot}$. We rank those nodes according to increasing t values ($t_0 = 0, .., t_S = T\text{-}Max$), and connect them with arcs $((Depot, t_i), (Depot, t_{i+1}))^{In}$

and $((Depot, t_i), (Depot, t_{i+1}))^{Out}$ accordingly. We also connect any node $(Depot, t)$ obtained this way to any existing node (x, u) such that $u = t + DIST_{Depot,x}$ and any existing node (x, u) to node $(Depot, t)$ such that $t = u + DIST_{x,Depot}$. We may consider the resulting network G^{*f} as a sub-network of $G_{T\text{-}Max}$, and its arcs e as provided with *carrier* costs CC_e as in the *VSR-Flow-Model*. Then we compute on G^{*f} a flow vector F, which satisfies (E7, E8, E9, E10) and which minimizes $\Sigma_{e \in ET\text{-}Max} F_e.Carrier\text{-}Cost_e$.

Remark 9. Step 6 always yields a feasible solution, since no non null *vehicle* flow value $f_{(x,t),(y,u)}$, $x \neq y$, is involved with $t < DIST_{Depot,x}$ or $(T\text{-}Max - u) < DIST_{y,Depot\ldots}$

7 NUMERICAL EXPERIMENTS

Purpose: Our purpose here is to:

- get a comparative evaluation of the lower bounds of Section 3;
- get a comparative evaluation of the 3 heuristic scheme described in Section 4, 5 and 6.
- test the influence of scaling coefficients α, β, δ and the impact of *preemption*.

Technical context: 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:

- Station set X is randomly generated as a 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 $DIST\text{-}S_{(x,y),(x',y')} = |x' - x| + |y' - y|$: $COST = \lambda.DIST$ or $COST = \lambda.DIST\text{-}S$;
- 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; That means that we allow here few *neutral* stations.
- *CAP* is randomly chosen between 10 and 20;
- *T-Max* is randomly chosen between = 30 and 100.

7.1 Testing the Impact of Scaling Coefficients α, β, δ

On a given instance $(X, v, CAP, T - Max, DIST, COST = DIST)$, we fix $\alpha = 10$, make vary β, δ with $\beta + \delta = 1$, and compute solutions through the *Shortest Cost/Distance* Strategy. We obtain

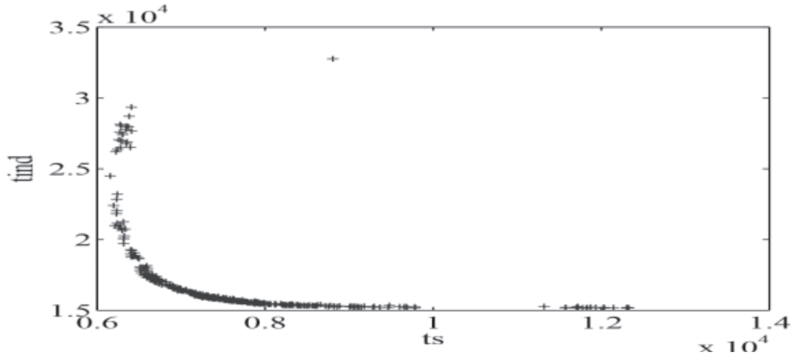

Figure 6 – Pareto frontier carrier riding cost *versus* vehicle riding time.

the Pareto frontier of Figure 6, with *ts* denoting the *carrier riding cost* and *tind* the *vehicle riding time*.

Comment: We see that *Carrier riding cost* and *vehicle riding time* behave like antagonistic criteria.

7.2 Comparing the Lower Bounds of Section 3

For several groups of 5 instances each related to a given size n, we compute the mean value of:

- *LB-Proj-Flow*: as defined in Theorem 5; *CPU-LB-Proj* denotes the related computing times in seconds.
- *LB-MCA* as defined in Theorem 3; *CPU-LB-MCA* denotes the related computing times in seconds.

When n is larger than 35, computing times for *LB-Proj-Flow* are too high. Still, we observe that CPLEX12 converges fast on the *LB-Proj-Flow* model, and so that imposing a threshold on CPU time is not likely to deteriorate the *LB-Proj-Flow* value in a significant way, even if it keeps us from mathematically proving that we get a lower bound this way. We get the following results (Symbol * means that we imposed a threshold of 1000 s on the running time for the *LB-Proj-Flow* model):

Comment: Experiments confirm both the better quality and the higher computing cost of the *LB-Proj-Flow* Lower bound.

7.3 Testing the Heuristics of Section 4, 5, 6 and the impact of Preemption

We compute, for the same groups of 5 instances as above, the average of the following *Global-Cost* values:

Table 1 – Lower Bounds with $\alpha = 10$, $\beta = 1$, $\delta = 0$.

n	*LB-Proj-Flow*	*CPU-LB-Proj*	*LB-MCA*	*CPU-LB-MCA*
20	84.8	46.6	82.3	8.6
30	96.5	3204.1	84.6	28.5
40	108.4	1000*	92.2	50.4
50	135.1	1000*	117.8	70.3
60	141.5	1000*	130.1	99.1

Table 2 – Lower Bounds with $\alpha = 10$, $\beta = 0$, $\delta = 1$.

n	*LB-Proj-Flow*	*CPU-LB-Proj*	*LB-MCA*	*CPU-LB-MCA*
20	182.7	67.6	176.9	6.5
30	228.2	1000*	216.2	25.7
40	235.6	1000*	218.7	52.6
50	299.9	1000*	288.3	75.4
60	297.3	1000*	270.1	109.0

- *SD*: obtained through *Shortest Cost/Distance Strategy* initialization of *VSR-MCA* $\Rightarrow$ *CPU-SD* is the related CPU time (s).
- *SD(50)*: obtained through *GRASP-VSR-MCA*, with $N = 1$ and $R = 50$.
- *LS(50)*: obtained through through *GRASP-VSR-MCA*, with $N = 50$ and $R = 1 \Rightarrow$ *CPU-LS* is the related CPU time.
- *VF*: obtained through the *Projected-Vehicle-Flow* heuristic $\Rightarrow$ *CPU-VF* is the related CPU time.
- *FRP*: obtained through application of the *Flow-Reconstruction-P-VSR* algorithm $\Rightarrow$ *CPU-FRP* is the related CPU time.
- *LB* denotes here the *LB-Proj-Flow* lower bound of the previous experiment.

We get (the computing time which were necessary in order to deal with the *LB-Proj-Flow* model re not taken into account in *CPU-VF* and *CPU-FRP*):

Comment: The improvement margin induced by the local search loop of the *VSR-MCA* algorithm is not very high, especially when the focus is on the *vehicle riding time*. A consequence is that performing random diversification through the use of the *replication* parameter R is most often more efficient. Both require small computational times. The *Projected-Vehicle Flow* algorithm provides similar results. At the end, the *Flow-Reconstruction-P-VSR* algorithm produces *preemptive* solutions whose *Global-Cost* value is always better that values obtained for the *Non*

Table 3 – Values SD, $SD(50)$, $LS(50)$, VF, FRP with $\alpha = 10, \beta = 1, \delta = 0$.

n	*LB*	*SD*	*CPU-SD*	*SD(50)*	*LS(50)*	*CPU-PI*	*VF*	*CPU-VF*	*FRP*	*CPU-FRP*
20	84.8	99.5	0.3	94.7	96.3	1.1	92.3	4.7	88.5	3.6
30	96.5	120.5	0.7	113.6	112.5	2.9	108.9	9.6	103.6	38.5
40	108.4	152.6	1.2	136.1	139.7	5.4	132.0	14.1	119.5	70.2
50	135.1	182.3	1.5	169.0	164.0	8.7	161.8	19.3	146.7	95.6
60	141.5	200.1	1.8	178.5	176.7	12.0	169.3	25.5	150.3	119.3

Table 4 – Values SD, $SD(50)$, $LS(50)$, VF, FRP with $\alpha = 10, \beta = 0, \delta = 1$.

n	*LB*	*SD*	*CPU-SD*	*SD(50)*	*LS(50)*	*CPU-PI*	*VF*	*CPU-VF*	*FRP*	*CPU-FRP*
20	182.7	220.0	0.4	212.1	217.6	1.4	205.6	5.8	191.0	3.9
30	228.2	273.1	0.8	264.6	270.9	3.3	255.7	10.2	241.5	33.5
40	235.6	297.5	1.3	277.7	288.7	6.0	264.6	14.9	247.3	64.0
50	299.9	372.2	1.5	346.3	364.7	9.5	335.9	21.0	312.0	120.3
60	297.3	378.4	1.9	348.9	369.8	11.8	340.8	28.4	321.9	122.7

Preemptive case. This should be amplified in case we allow more *neutral* stations. We may extrapolate that, on our instances, lower bound *LB-Proj-Flow* probably misses the optimal value of *Non Preemptive VSR* problem by about 10%, and is close to optimal value of *Preemptive VSR*. Finally, we must notice that a limitation for both *Flow-Reconstruction-P-VSR* and *Projected-Vehicle-Flow* is that they rely on the resolution of *Projected-VSR-Flow* instances defined on an almost complete oriented graph, whose computing costs increase fast with the number n of stations.

8 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 *Network Flow* approaches which puts the focus on the way *vehicles* move from *excess* stations to *deficit* ones. Still, many open problems remain, related to the design of exact algorithms and also, if we refer to practical context, to the way algorithms which have been designed for static models 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] Anily S, Gendreau M & Laporte G. 2011. The uncapacitated swapping problem on line and circle. *Networks*, **58**: 83–94.

[2] Archetti C & Speranza M. 2008. The split delivery vehicle routing problem, a survey; in The vehicle routing problem: latest adances and new challenges; p. 103–122, Springer U.S.

[3] ARONSON J. 1989. A survey on dynamic network flows. *Annals of O.R*, **20**: 1–66.

[4] BARTH M & TODD M. 1999. Simulation model performance analysis of a multiple station shared vehicle system. *Transportation Research Part C: Emerging Technologies*, **7**(4): 237–259.

[5] BENCHIMOL M DE P, CHAPPERT B, TAILLE A DE LA, LAROCHE F, MEUNIER F & ROBINET L. 2011. Balancing the stations of a self service bike hiring systems. *RAIRO-RO*, **45**: 37–61.

[6] BORDENAVE C, GENDREAU M & LAPORTE G. 2012. A branch and cut algorithm for the preemptive swapping problem. *Networks*, **59**(4): 387–399.

[7] BOYACI B, ZOGRAFOS K & GEROLIMINIS N. 2015. An optimization framework for the development of efficient one-way car-sharing systems. *EJOR*, **240**(3): 718–733.

[8] CARLIER A, MUNIER-KORDON A & KLAUDEL W. 2015. Mathematical model for the study of relocation strategies in one-way carsharing systems. *Transportation Research Procedia*, **10**: 374–383.

[9] CARLIER A. 2016. PhD PARIS VI University, France.

[10] CHEMLA D, MEUNIER F & CALVO RW. 2013. Bike sharing systems: solving the static rebalancing problem. *Discrete Optimization*, **10**(2): 120–146.

[11] CONTARDO C, MORENCY C & ROUSSEAU LM. 2012. Balancing a dynamic public bike-sharing system. *Rapport CIRRELT*, Univ. MONTREAL.

[12] CORDEAU JF & LAPORTE G. 2003. A Tabu search heuristic algorithm for the static multi-vehicle Dial and Ride Problem. *Transportation Research B*, **37**: 579–594.

[13] CORNUEJOLS G, NEMHAUSER GL & WOLSEY LA. 1990. "The uncapacited facility location problem". In: MIRCHANDANI PB & FRANCIS RL (Eds.), Discrete Location Theory, p. 119–171.

[14] GAVALAS D, KONSTANTOPOULOS C & PANTZIOU G. 2015. Design and management of vehicle sharing systems: a survey of algorithmic approaches; ArXiv e-prints, arxiv.org/pdf/1510.01158, October.

[15] GHIANI G, GUERRERO F & MUSMANNO F. 2002. "The capacitated plant location problem with multiple facilities in the same site". *Computer and Operations Research*, **29-13**: 1903–1912.

[16] HANSEN P, BRIMBERG J, UROSEVIC D & MLADENOVIC N. 2007. "Primal-dual local search for plant location". *INFORMS Jour. Comp.*, **19-4**: 58–72.

[17] PEREZ HH & GONZALEZ JS. 2004. Heuristics for the one commodity pick up and delivery traveling salesman problem. *Transportation Sciences*, **38**: 244–255.

[18] KEK GH, CHEU RL, MENG Q & FUNG C.HA. 2009. A decision support system for vehicle relocation operations in carsharing. *Transport. Res. E: Logistics/Transp. Review*, **45**(1): 149–158.

[19] KLOSE A & DREXL A. 2005. "Facility location models for distribution systems". *EJOR*, **162**: 429–449.

[20] LEE J & PARK GL. 2013. Design of a team based relocation scheme in electric vehicle sharing systems. *Proc. Int. Conf. Computational Sciences and Applications*, **7973**: 368–377.

[21] LEE J, PARK GL, KANG MJ, KIM J, KIM HJ, KIM LK & KO YI. 2012. Design of an efficient matching based relocation scheme for electric vehicle sharing systems. *Proc. Int. Conf. in Computer Applications for Modeling, Simulation and Automobile*, **341**: 109–115.

[22] MIRCHANDANI PB & FRANCIS RL. 1990. Discrete Location Theory. J. Wiley Sons.

[23] NOURINEDJAD M & ROORDA MJ. 2014. A dynamic car sharing decision support system. *Transportation Research E*, **66**: 36–50.

[24] RAINER-HARBACH M, PAPAZECK P, HU B & RAIDL G. 2015. Balancing bicycle sharing systems: a variable neighbourhood search approach. *Journal of Global optimization*, **63**(3): 597–629.

[25] RAVIV T, TZUR M & FORMA I. 2013. Static repositioning in a bike sharing system: models and solution approaches. *EURO Journal on Transportation and Logistics*, **2**(3): 187–229.

[26] RESENDE M & WERNECK R. 2004. "Hybrid heuristic for p-median problem". *J. Heuristics*, **10**: 59–88.

[27] SAYARSHAD H, TAVASSOLI S & ZHAO F. 2012. A multi-periodic optimization formulation for bike planning and bike utilization. *Applied Mathematical Modelling*, **36**(10): 4944–4951.

[28] SHAHEEN SA, GUZMAN S & ZHANG H. 2010. Bike sharing in Europe, the Americas and Asia. *Transportation Res. Record: Journal Transportation Research Board*, **2143**(1): 159–167.

[29] WASERHOLE A, JOST V & BRAUNER N. 2012. Vehicle sharing system pricing regulation, http://hal.archivesouvertes.fr/docs/00/82/81/92/PDF/.

[30] WHITAKER R. 1983. "A fast algorithm for the greedy interchange of large scale clustering and median location problems". *INFOR*, **21**: 95–108.