CASPT 2015

Constraint Propagation for the Dial-a-Ride Problem with Transfers

Samuel Deleplanque(Speaker)[1,2], Alain Quilliot[2]

Lab': 1-ULB (Belgium); 2-LIMOS (France)

Outline

1. Definitions and model
2. State of the Art
3. Heuristics based on Constraint Propagation:
   - DARP
   - DARPT
4. Experiments

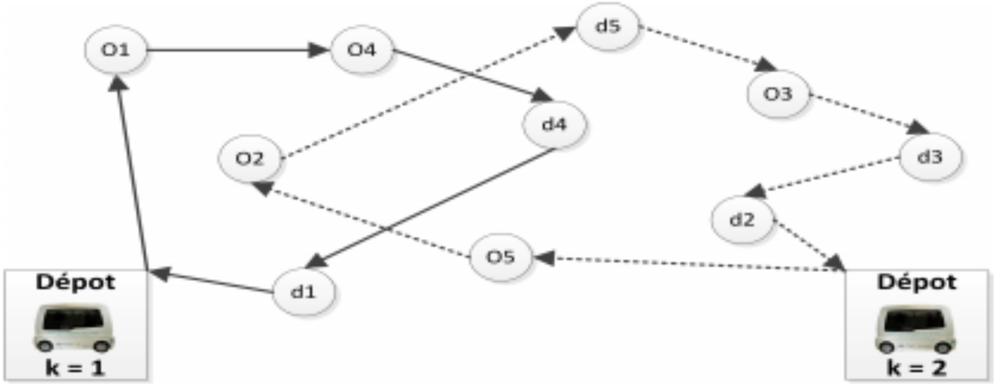
Static DARP

An instance is defined by:

- a fleet VH of K vehicles (with a capacity *CAP* and a maximum route time);
- a demand set: $D = (D_i = (o_i, d_i, \Delta_i, F(o_i), F(d_i), Q_i), i \in I)$
  - (the origin node, the destination node, maximum ride time, two time windows, the load resp.);

And the related graph G = (V, E), which contains:

- the 2*|K| Depot nodes,
- the origin and destination nodes of the demands.
- the arcs e in E endowed with riding times l(e) ≥ 0;

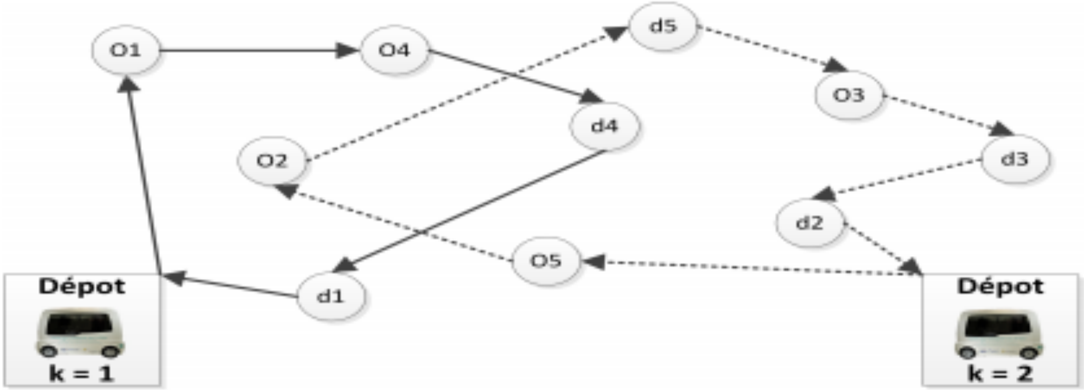


Static DARP
Focus on the Time Constraints

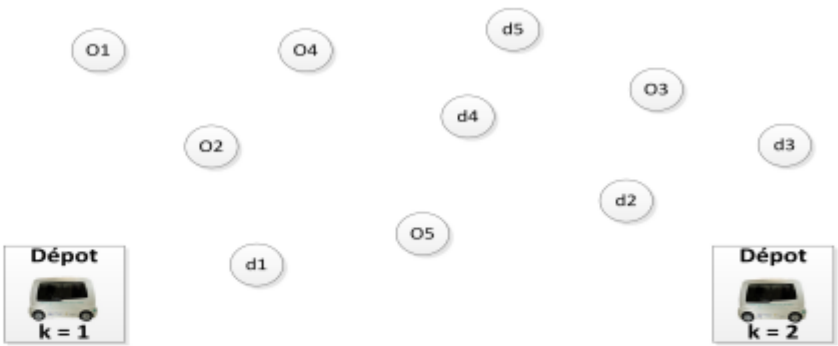
3 sets of time constraints:

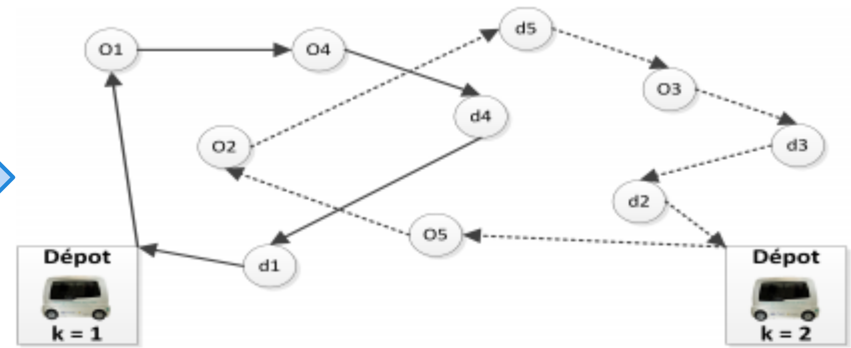
- time windows (from the demands),
  - on the origin and destination nodes;
- maximum ride time (from the demands),
- maximum route time (from vehicles).

=> All of these constraints will be “handle” by constraint propagation.

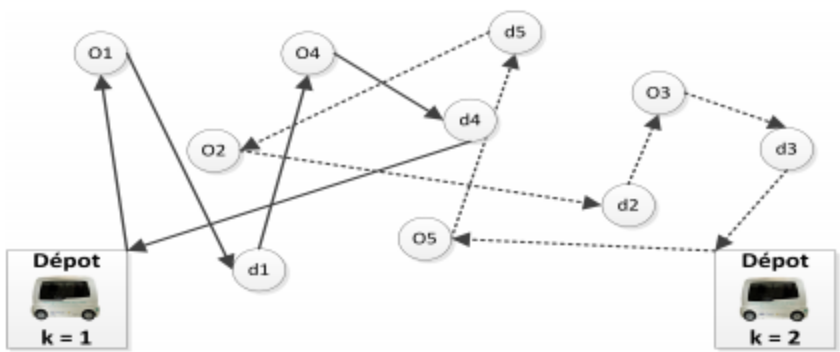


Performance criteria





Min(global duration of the tours)



Min
(ride times)

multi-criteria

short state of the art

DARP

Tabu search - A tabu search heuristic algorithm for the static multi-vehicle dial-a-ride problem - J.-F. Cordeau et al., - Transportation Research – 2003 ;

<u>Insertion techniques</u> (IT)

A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem - J. Jaw et al. - Transportation Research – 1986.

DARPT (rare...)

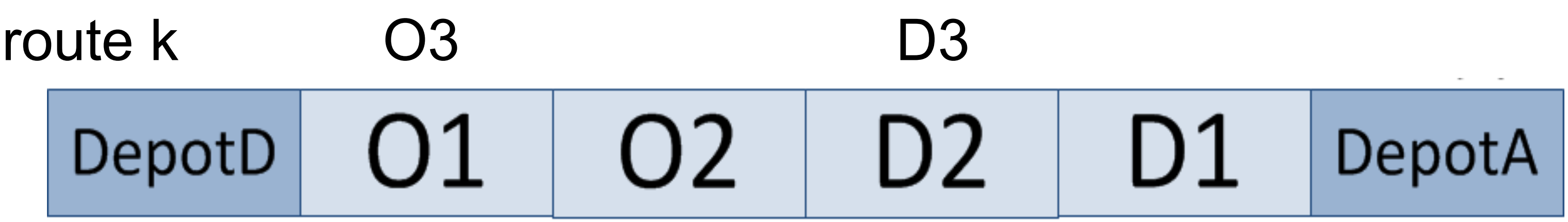
Adaptive Large Neighborhood Search - Masson, Renaud, Fabien Lehuédé, and Olivier Péton. "The dial-a-ride problem with transfers." Computers & Operations Research 41 (2014): 12-23.

PDPTWT

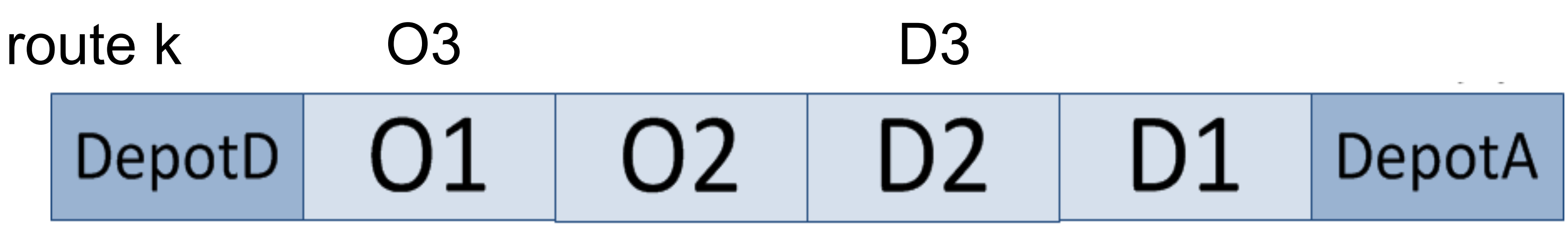
Branch-and-Cut - **<u>Insertion techniques</u>** - **VNS**

Heuristic based on constraint propagation
Main ideas: the insertions

- Demand insertions: <u>one after another</u>
  - selection of the demand according to the number of cars available for the specific demand (i.e. without violation of constraints):
    - => Random selection among the best candidates set,
  - selection of the insertion parameters according to the smaller impact on the total route cost:
    - => Random selection among the best candidates set.
- Example of insertion parameters: (k, O1, D2) for inserting the demand 3:

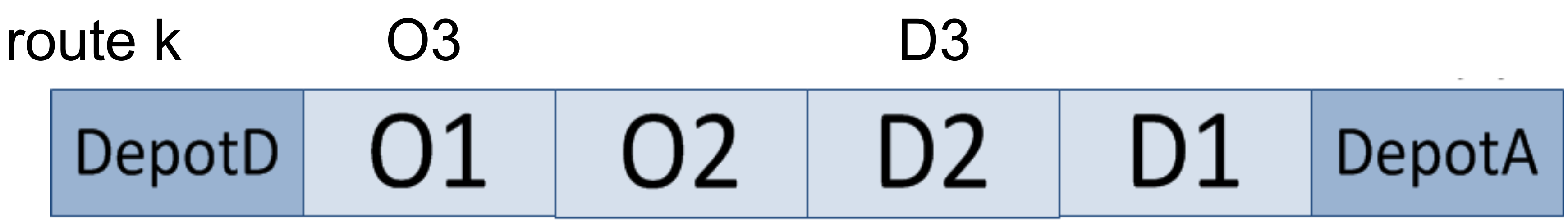


- For a given state of the routes, we check the feasibility and evaluate each insertion possible:
    - by constraint propagation of the three types of time constraints;
- After an insertion:
    - Update of the insertion parameters for the modified route k,
    - Add new parameters for k according to the new nodes in the route.



Heuristic based on constraint propagation
Main ideas - Monte Carlo Process

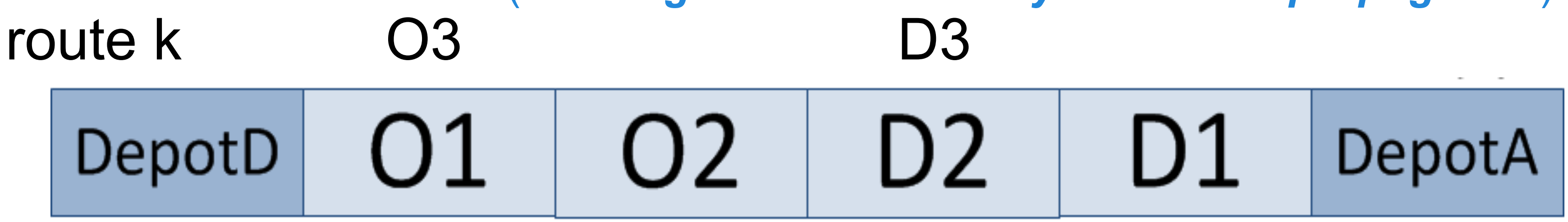
- We use a random number generator for:
  - the selection of the “best” demands,
  - the selection of the “best” insertion parameters;
- We launch several replications of our heuristic using the same generator:
  - The process can be stopped once a first solution is obtained,
  - The process can be paralyzed.



Simple Insertion heuristic

Initialize the *K* routes with depot nodes (departure & arrival)
For all *j* in *D*, **FREE(j) <- all the possible 4-uple (k, O, D, v)**
While *D* not empty
    pick up some demand *i* in *D* on a set of N1 demands with the lower number of vehicle available
    if *FREE(i)* is *Nil* reject *i*
    else
        select 4-uple *(k, O, D, v)* according to the N2 best *v*
        insert *i* in *k* by *D* and *D* and remove the 4-uple
        upgrade and update the 4-uples related to *k* in the *FREE* sets

(***creating the sets FREE by constraint propagation***)



(time) Constraint Propagation Inference Rules

$$y = Succ(\Gamma, x);\ \mathcal{FS}.min(x) + DIST(x, y) > \mathcal{FS}.min(y)$$
$$\models$$
$$\mathcal{FS}.min(y) \leftarrow \mathcal{FS}.min(x) + DIST(x, y);\ NFact \leftarrow y$$

R1

$$y = Succ(\Gamma, x);\ \mathcal{FS}.max(y) - DIST(x, y) < \mathcal{FS}.max(x)$$
$$\models$$
$$\mathcal{FS}.max(x) \leftarrow \mathcal{FS}.max(y) - DIST(x, y);\ NFact \leftarrow x$$

R2

$$y = Twin(x);\ x \ll_\Gamma y;\ \mathcal{FS}.max(y) > \mathcal{FS}.max(x) + \Delta(x)$$
$$\models$$
$$\mathcal{FS}.max(y) \leftarrow \mathcal{FS}.max(x) + \Delta(x);\ NFact \leftarrow y$$

R3

$$y = Twin(x);\ x \ll_\Gamma y;\ \mathcal{FS}.min(x) < \mathcal{FS}.min(y) - \Delta(x)$$
$$\models$$
$$\mathcal{FS}.min(x) \leftarrow \mathcal{FS}.min(y) - \Delta(x);\ NFact \leftarrow x$$

R4

$$x \in \Gamma;\ \mathcal{FS}.min(x) > \mathcal{FS}.max(x)$$
$$\models$$
$$\mathit{Fail}$$

R5

Extended constraint Propagation for testing an insertion (DARPT)

Testing a transfer => Add the new precedence constraints in the set of inference rules

$$y = \mathit{Twin}(x);\ \mathit{Status}(x) = \mathit{Out\text{-}Reload};\ \mathcal{FS}.\min(x) > \mathcal{FS}.\min(y)$$

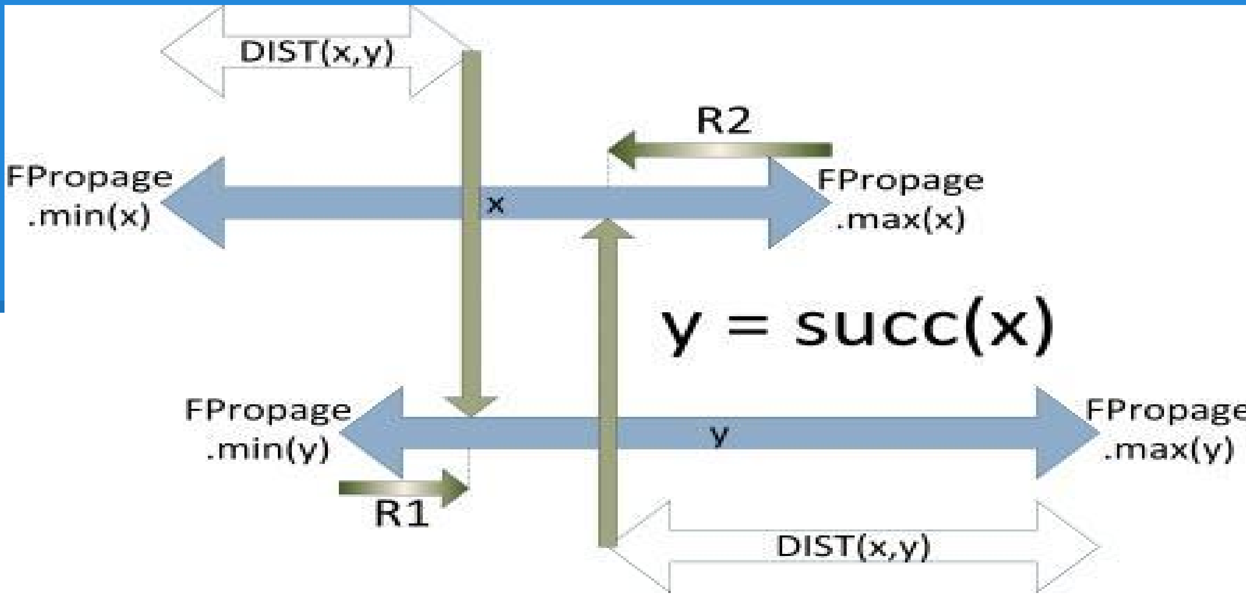
$$\models \quad \text{R6}$$

$$\mathcal{FS}.\min(y) \leftarrow \mathcal{FS}.\min(x);\ \text{NFact} \leftarrow y$$

$$y = \mathit{Twin}(x);\ \mathit{Status}(x) = \mathit{Out\text{-}Reload};\ \mathcal{FS}.\max(x) > \mathcal{FS}.\max(y)$$

$$\models \quad \text{R7}$$

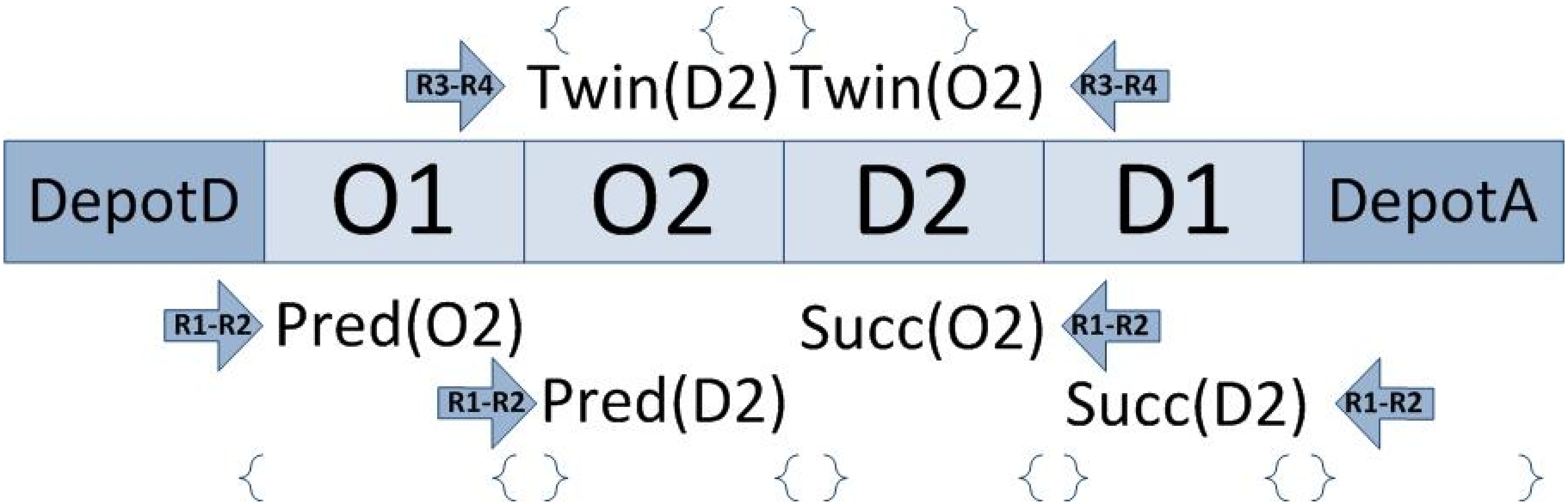
$$\mathcal{FS}.\max(x) \leftarrow \mathcal{FS}.\max(y);\ \text{NFact} \leftarrow x$$



y = Succ(x) :

e.g. R1:

FPropage.Min(x) + DIST(x,y) <= FPropage.Min(y)



Experiments on the DARP

Inst.	Lb	Opti	cpu*	TI	Gap	cpu
a2-16	294,25	294,25	1	294,25	0,00	0
a2-20	344,83	344,83	3	344,83	0,00	0
a2-24	431,12	431,12	9	431,12	0,00	0
a3-18	300,48	300,48	5	300,81	0,11	1
a3-24	344,83	344,83	8	344,83	0,00	2
a3-30	494,85	494,85	10	495,26	0,08	16
a3-36	583,19	583,19	105	589,86	1,14	14
a4-16	282,68	282,68	6	283,10	0,15	0
a4-24	375,02	375,02	6	376,21	0,32	94
a4-32	485,50	485,50	31	487,10	0,33	29
a4-40	557,69	557,69	8328	565,95	1,48	63
a4-48	668,82	NA	14543	700,30	NA	31

Inst. : aK-|D|

Perf: Min Total Distances

Gap < 2% and CPU(s) < 100s

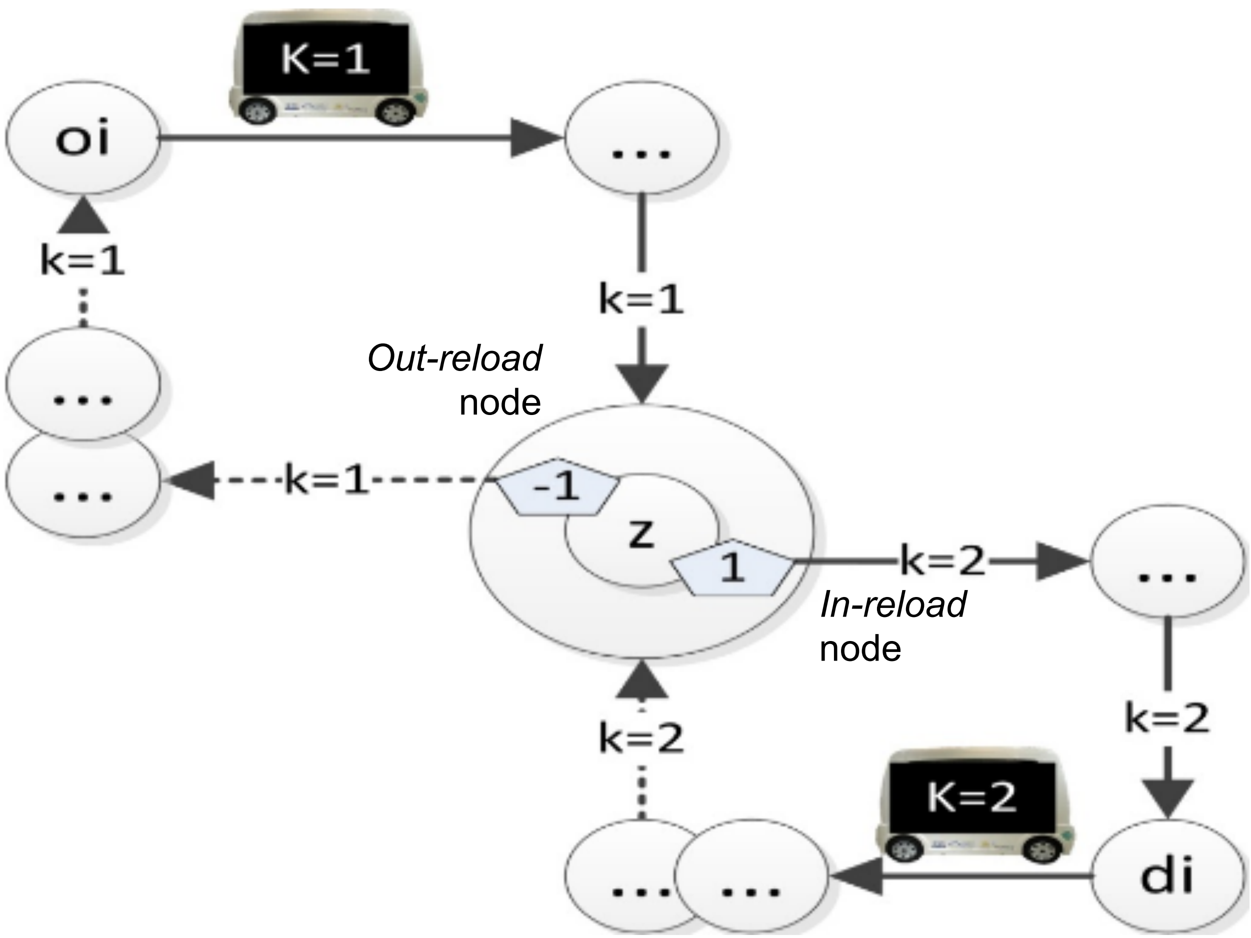
Less Good on very tight instances (Cordeau 2003)

(=> needs a local search)

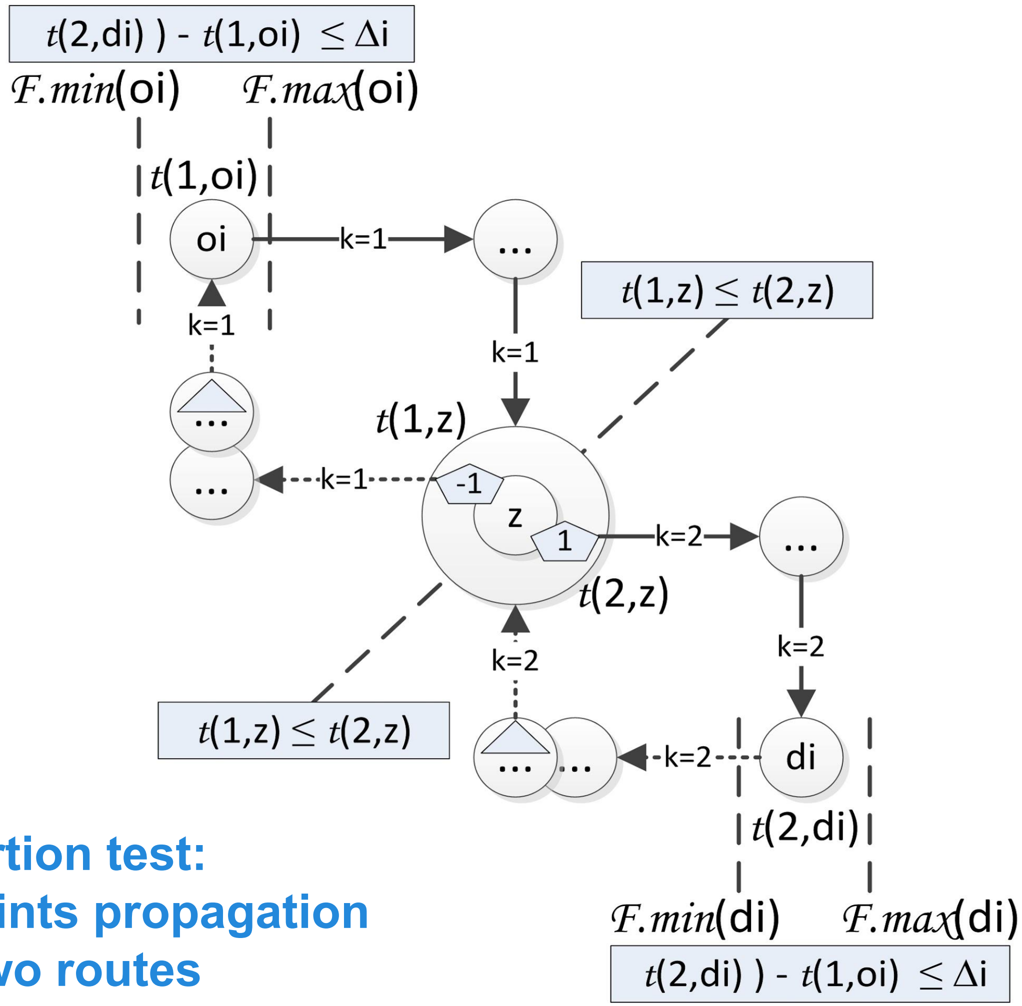
J-F Cordeau. A Branch-and-cut Algorithm for the Dial-a-ride. Operations Research May/June, p573-586. 2006.

S. Parragh. Introducing heterogeneous users and vehicles into models and algorithms for the dial-a-ride problem. Transportation Research Part C : Emerging Technologies. Volume 19, Issue 5, p912-930. 2011.

Transfer allowed ("anywhere")

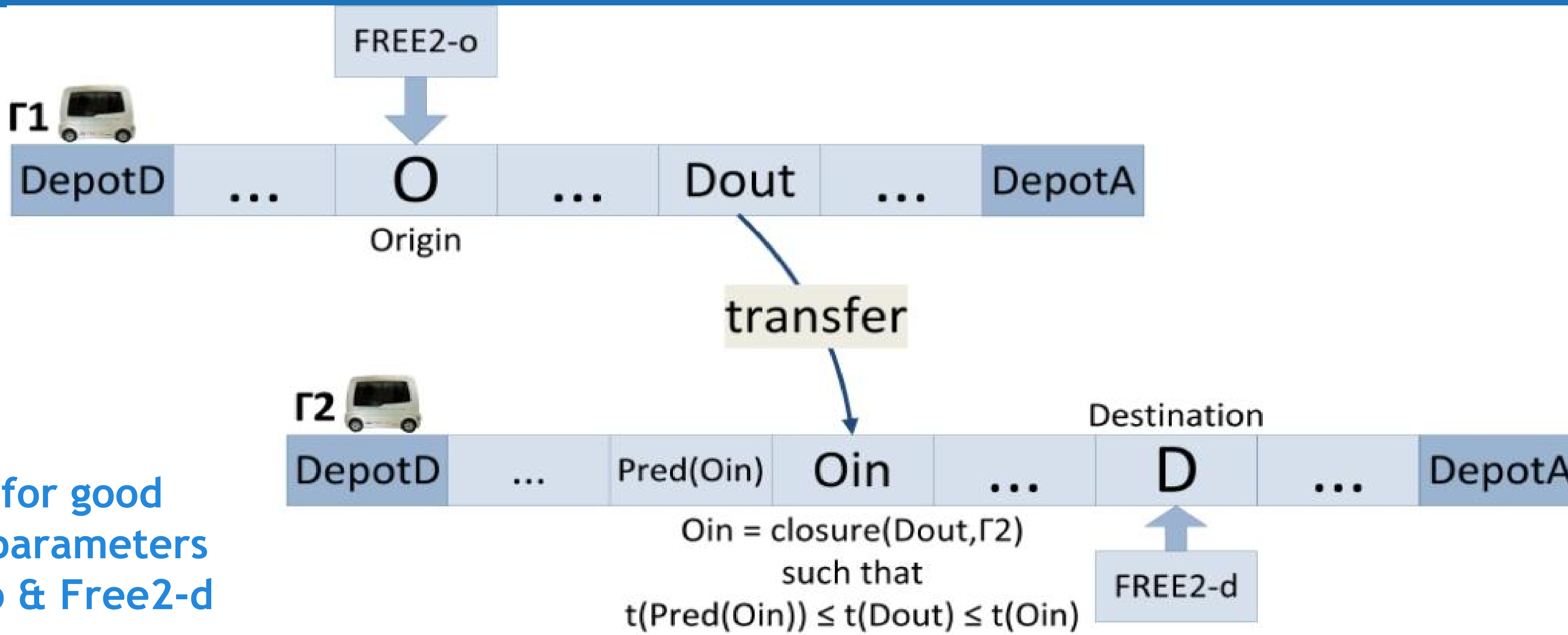


New time constraints



For each insertion test:
(time) constraints propagation
(at least) on two routes

Creating the FREE2 set: locating the transfer node



Searching for good insertion parameters

1. Free2-o & Free2-d (easy)
2. FREE2 with Free2-o & Free2-d, searching for (Dout,Oin) and transfer node.

Search a node Dout in the route Γ1, such that:

- O << Dout
- *Closure*(Dout, Γ2) << D;
- DIST(Dout, *Closure*(Dout, Γ2) is the smallest possible

FREE2 <- (i, Γ1, Γ2, O, Dout, Oin = *Closure*(Dout, Γ2), D, Transfer Node)

The General Algorithm

When should we use the transfers?

- INSERTION1 and INSERTION2 solve the DARP and the DARPT, respectively.
- Δ-Aux Current maximum ride time

Algorithm

Δ-Aux <- Δ; Initialize λ with a large value Λ;

For p = 1..P do

  Δ <- *Update-Δ* (λ, Δ);

  (T1, t1, *Perf1*, *Reject1*) <- INSERTION1(N_1, N_2);

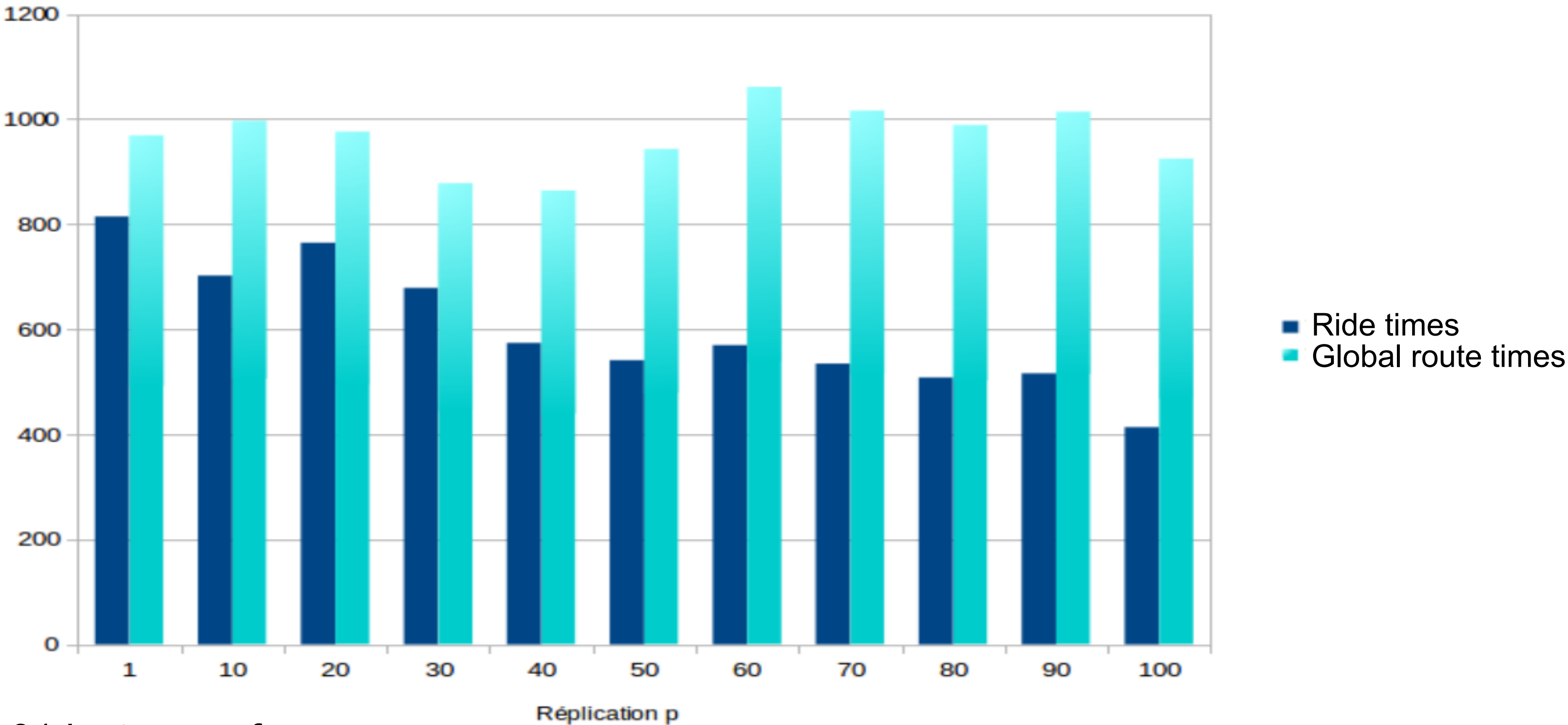
  If *Reject1* = Nil then (T, t, *Perf*, *Reject*) <- (T1, t1, *Perf1*, *Reject1*)

  Else (T, t, *Perf*, *Reject*) <- INSERTION2(T1, t1, *Reject1*, N_7);

  Δ <- Δ-Aux ; Update λ: λ <- λ – 1/P.(Λ – 1);

Keep the best result (T, t, *Reject*, $Perf_{A, B, C}$(T, t)) which was obtained during this process.

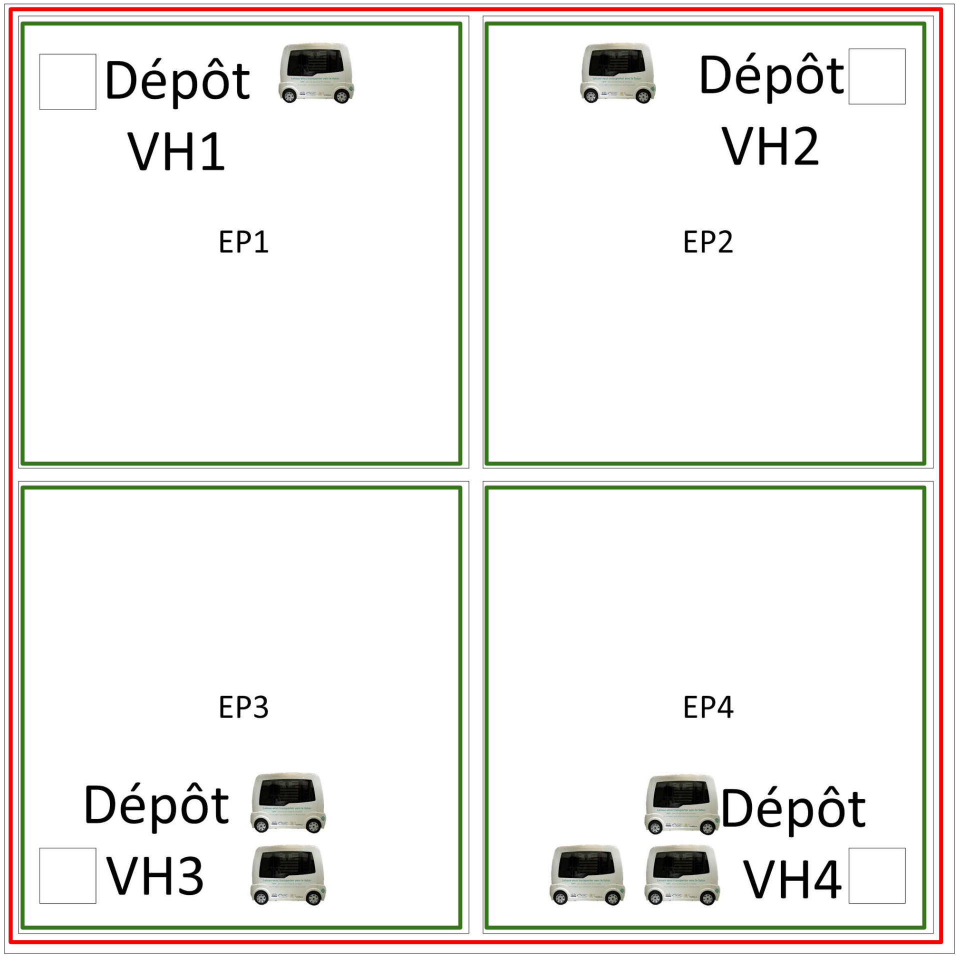
Ride time evolution on a simple instance



Pr01 instance of:

Cordeau, J.-F. and Laporte, G. (2003). A tabu search heuristic algorithm for the static multi-vehicle dial-a-ride problem. *Transportation Research B 37*, 579–594.

Experimentations - DARPT Clustering



VHi : « sub-platoon »

EPi : « sub-space »

30% local demands

70% general demands

$$\Delta_i = \beta DIST(o_i, d_i), i = 1..|D|, \beta \geq 1$$

$$F_{o_{i_{aller}}} = [690 + g; 690 + g + 10]$$

$$F_{d_{i_{aller}}} = [690 + g; 690 + g + \Delta_i]$$

$$F_{o_{i_{retour}}} = [840 - g - \Delta_i; 840 - g]$$

$$F_{d_{i_{retour}}} = [840 - g - 10; 840 - g]$$

Experiments - Results
DARP Vs DARPT

$\|D\|$	K	CAP	α	β	EF	$Insert$	$Insert_{Tr}$	Gap
32	4	6	0,3	20	5	38,80	52,73	35,89
32	4	6	0,3	20	15	54,59	62,76	14,95
32	4	6	0,3	20	30	61,84	68,32	10,47
32	5	6	0,3	20	5	49,74	70,68	42,08
32	5	6	0,3	20	15	70,28	86,02	22,40
32	5	6	0,3	20	30	77,00	89,07	15,67
64	4	6	0,3	20	5	21,15	28,48	34,66
64	4	6	0,3	20	15	29,37	34,34	16,94
64	4	6	0,3	20	30	35,29	37,05	4,97
64	5	6	0,3	20	5	27,60	38,68	40,14
64	5	6	0,3	20	15	39,06	46,12	18,08
64	5	6	0,3	20	30	45,17	48,20	6,72
96	4	6	0,3	20	5	15,72	20,90	33,00
96	4	6	0,3	20	15	22,43	24,16	7,74
96	4	6	0,3	20	30	25,92	27,02	4,24
96	5	6	0,3	20	5	20,48	28,24	37,89
96	5	6	0,3	20	15	28,64	32,62	13,93
96	5	6	0,3	20	30	33,26	35,35	6,30

18 set of 5 instances

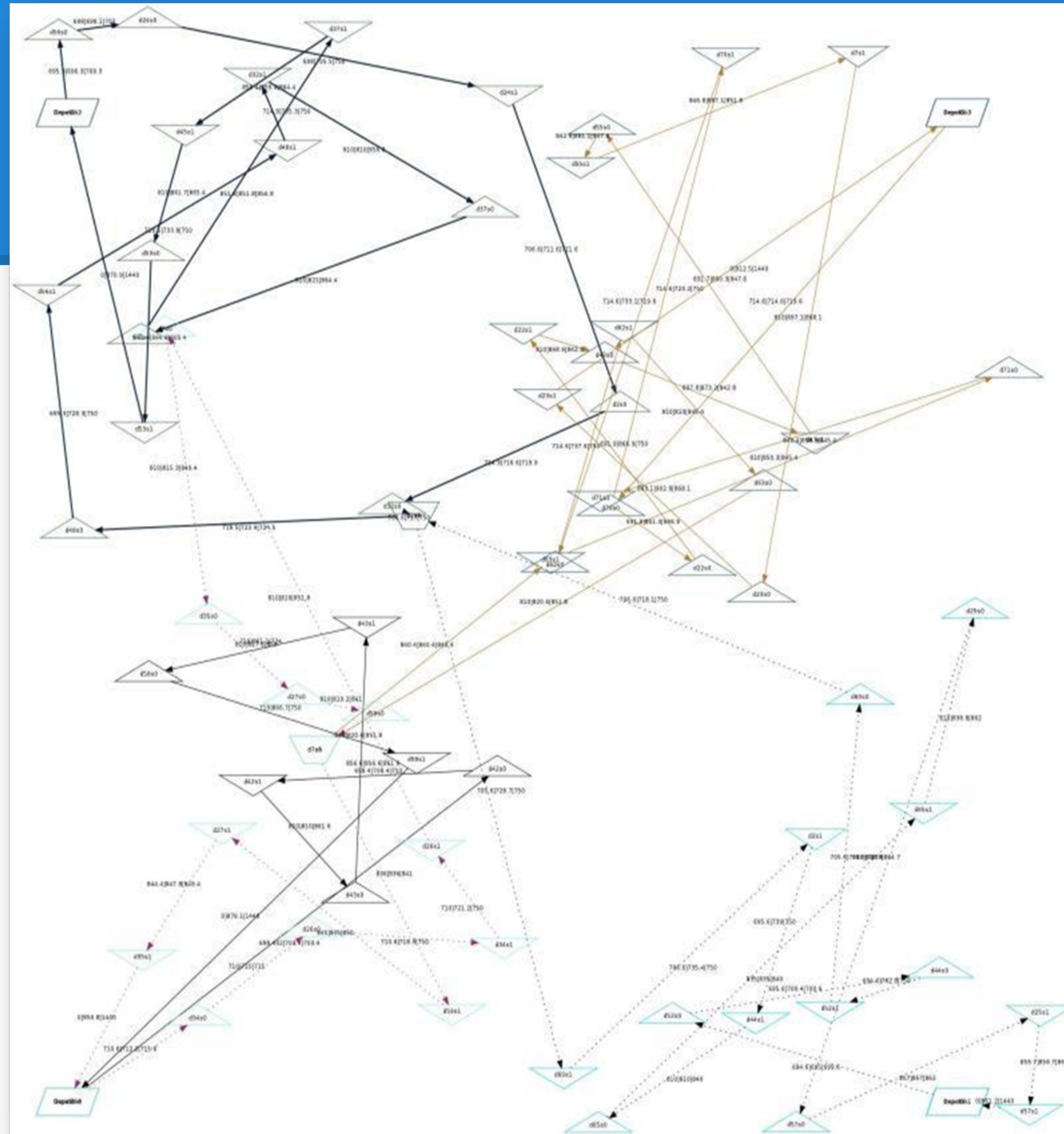
EF: *F.Max(x) - F.Min(x), x origin or destination*

Insert, $Insert_t$: Rates of insertions

Gap: 100.($Insert_t$- *Insert*)/(*Insert*)

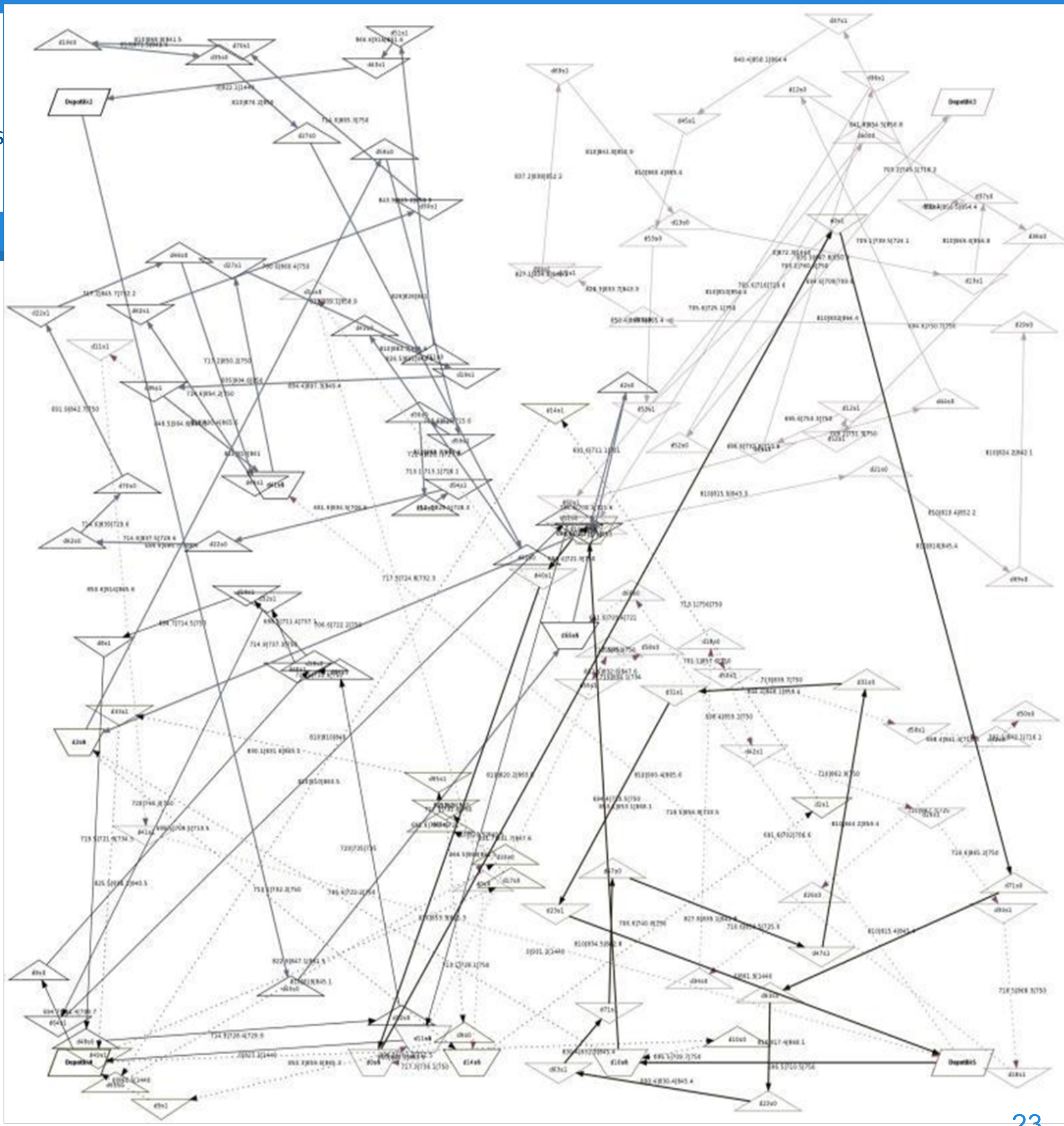
Visualization of a solution

- small instance



Visualization of a solution

- medium instance



CASPT 2015

Thank you!

Any questions?

Samuel Deleplanque(Speaker)[1,2], Alain Quilliot[2] , Lab': 1-ULB (Belgium); 2-LIMOS (France)

Robustness Tools in dynamic DARP. S. Deleplanque, A. Quilliot, In Recent Advances in Computational Optimization. Studies in Computational Intelligence, Vol. 580, 2015, pp 35-51, Springer

Constraint Propagation for the Dial-a-Ride Problem with Split Loads. S. Deleplanque, A. Quilliot, In Recent Advances in Computational Optimization. Studies in Computational Intelligence, Vol. 470. Springer