Multiperiodic VRP models and hybrid OR-CP solution techniques For closed-loop reverse logistics

Pierre DEJAX
Emilie GRELLIER, Narendra JUSSIEN
Ecole des Mines de Nantes
IRCCyN - LINA

VIP'08, OSLO
June 12-14, 2008

ECOLE DES MINES DE NANTES

LABORATOIRE D'INFORMATIQUE DE NANTES ATLANTIQUE

Plan

1. **Context and problem description**

2. Metaheuristics : GRASP

3. Column Generation / Branch and Bound

4 Results

5. Conclusions and perspectives

Reverse Logistics

open loop / closed loop supply chains ?

"Reverse logistics can be viewed as the process of planning, implementing and controlling the efficient, cost effective flow of raw materials, in-process inventory, finished goods and related information, from the point of consumption to the point of origin, for the purpose of recapturing their value or proper disposal".

American Reverse Logistics Executive Council

(Rogers and Tibben-Lembke, 1998)

Motivation :

- Environmental protection, product, packages recycling
- Productivity and cost savings : product repair, handling equipements reuse
- Customer service : product returns

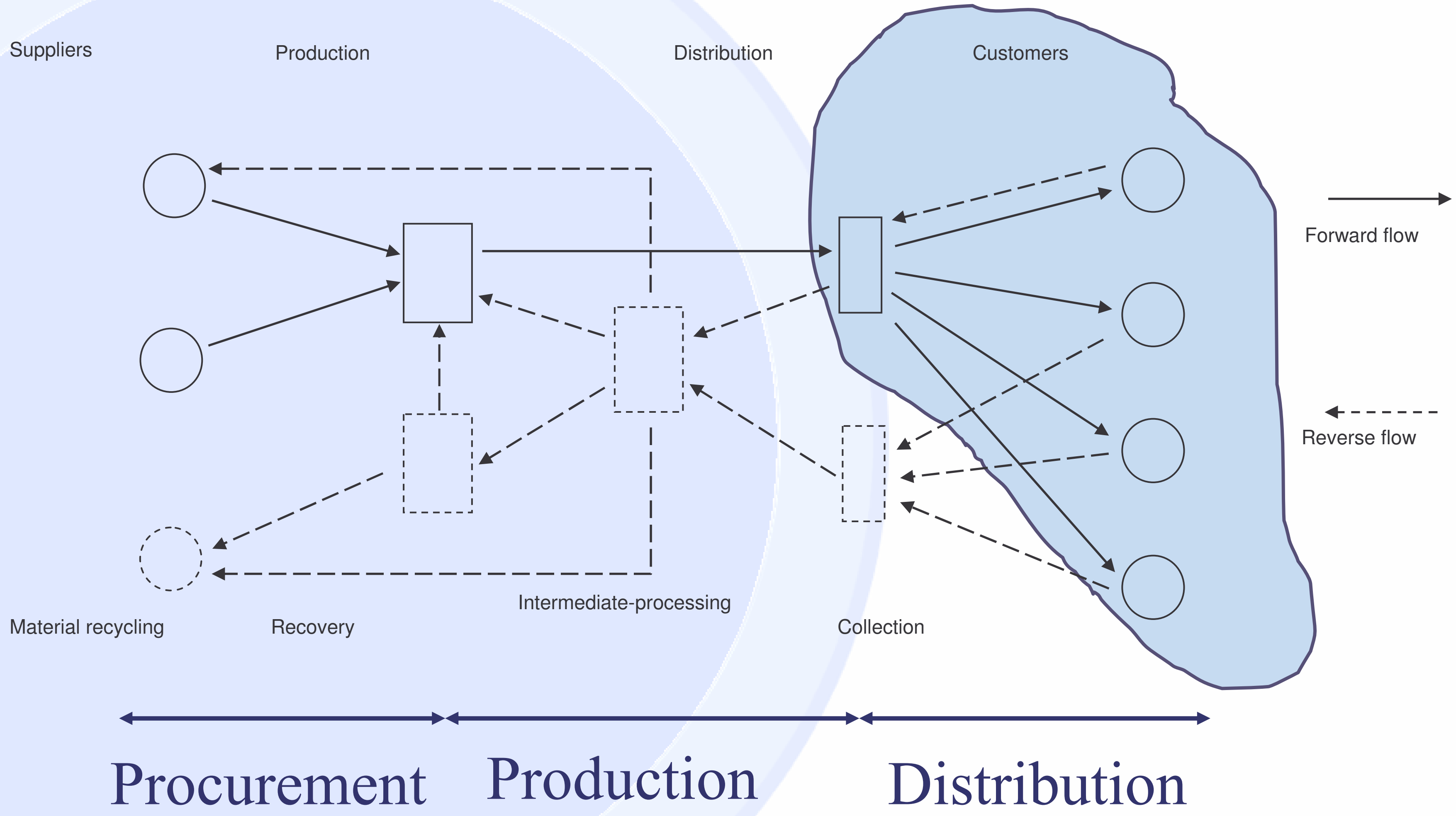
General References, books and surveys

- Fleischmann, M., J. M. Bloemhof-Ruwaard, R. Dekker, E. Van Der Laan, J. A.E.E. Van Nunen, L. N. Van Wassenhove, 1997. Quantitative models for reverse logistics: a review.
- Gungor, A., S.M. Gupta, 1999. Issues in Environmentally Conscious Manufacturing and Product Recovery: A Survey.
- Fleischmann, M., 2001. Quantitative models for reverse logistics.
- Dekker R., Fleismann M., Interfuth K., van Wassenhove L edts., 2004, "Reverse logistics. Quantitative models for closed loops supply chains",436 p.
- Bostel N., P. Dejax, Z. Lu, 2005."The Design, Planning and Optimization of Reverse Logistic Networks"
- CAOR, 2006. Special issue on Reverse Logistics.
- Stock J.R., 2008. "Reverse Logistics, Green Logistics and Packaging".

Case studies based models

- Thierry, M.C., M. Salomon, J. Van Nunen, L. Van Wassenhove, 1995. Strategic issues in product recovery management.
- De Brito, M. P., R. Dekker, S. D.P. Flapper, 2003. Reverse Logistics – a review of case studies.
- Kumar S. and P. Malegeant, 2006. Strategic alliance in a closed-loop supply chain, a case of manufacturer and eco-non-profit organization.
- Schultmann F., Zumkeller M. and O. Rentz, 2006. Modeling reverse logistic tasks within closed-loop supply chains: An example from the automotive industry.
- Schultmann F., Engels B. and Rentz O., 2004. Flowsheeting-based simulation of recycling concepts in the metal industry.
- Seitz M., 2007, A critical assessment of motives for product recovery: the case of engine remanufacturing.
- Krikke H., le Blanc I., van Krieken M., Fleuren H., 2008. Low-frequency collection of materials disassembled from end-of-life vehicles: On the value of on-line monitoring in optimizing route planning.

Integrated supply chain with forward and reverse flows (closed loop)

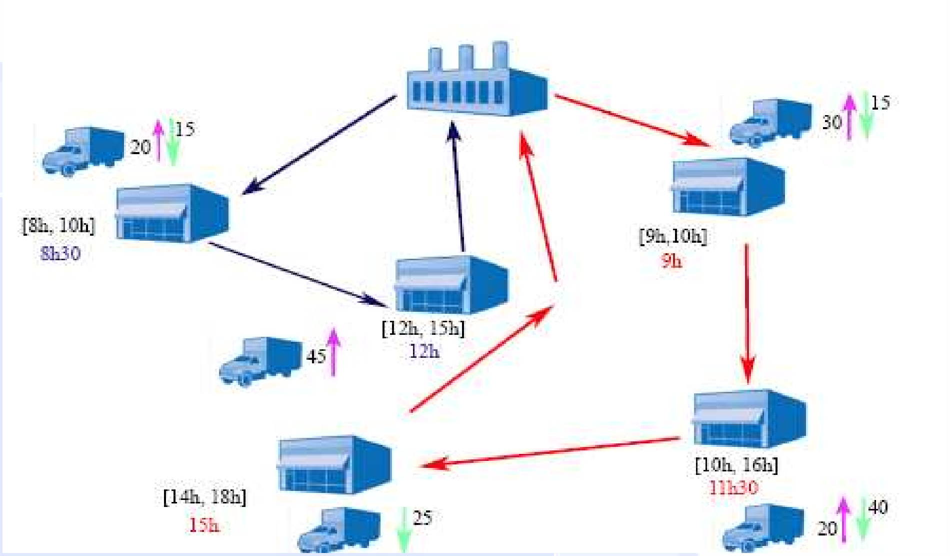


Goal of the research

- Study a « generic » realistic distribution system
  - multiperiodic VRP with reverse flows
  - several possible management policies

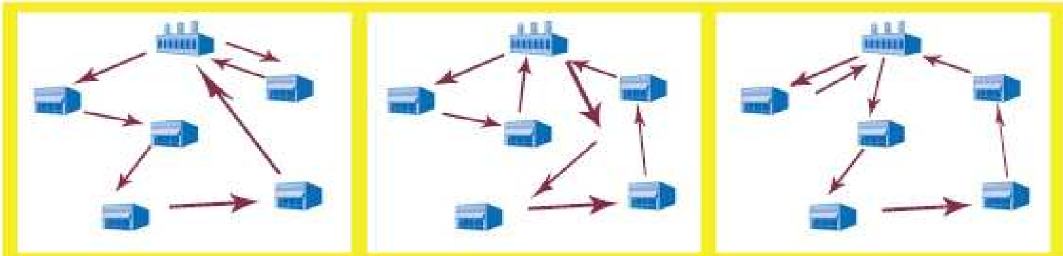
- Evaluate different solution technique:
  - pure O.R. techniques,
  - Hybrid O.R. techniques with C.P. techniques
  - Classical heuristics and meta heuristics: GRASP
  - « exact » methods : Column generation/ Branch & Bound

VRP with Reverse Flows

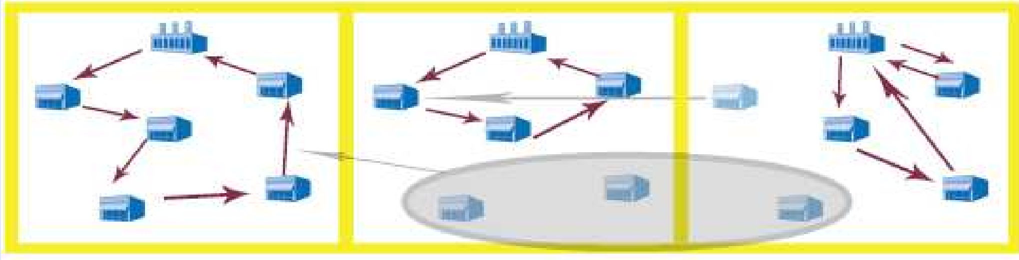


- 1 warehouse,
- n stores
- 1 product type on pallets
- pallets to be recycled
- products returnable by consumers
- demand on a multiperiodic horizon
- dtore time windows
- homogeneous fleet of vehicles

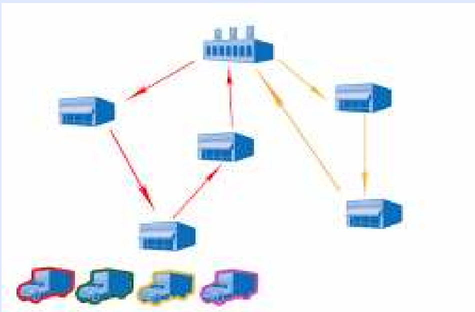
Possible management policies:



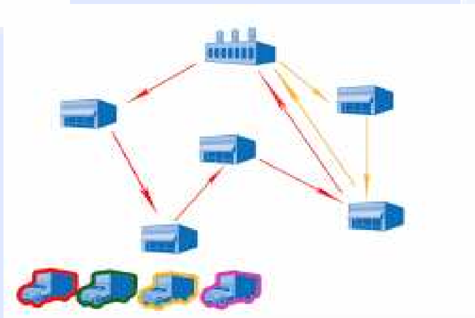
•Just in time



•Storage



•Full delivery



•Split delivery

General goals

- demand satisfaction
- handling of returns
- respect vehicle capacities
- Minimize costs :
  - -Vehicle fixed and transport costs
  - -Possible storage costs at store

Related VRP problems and methods

- VRP: Toth &Vigo (2001)
- VRPTW: Cordeau et al.(2002), Braÿsy & Gendreau (2005)

- Pick up & Delivery: Cambpell et al. (2002), Nagy & Saly (2004), Parragh et al. (2008)
  - Simultaneous Delivery & Pick up: Dethloff (2001), Halskau et al. (2001), Hoff & Lokketangen (2006)

- Inventory Routing: Deseaulnier et al. (2002)
  - IPDPTW: Christiansen & Nygreen (1998), Christiansen (1999) *[planning of ships between ports over a month]*

- Hybrid methods for routing problems:
  - Rousseau et al.(2002): constraint programming, column generation
  - Caseau & Laburthe (1999): solving large VRPs with insertion methods using LDS

Plan

1. Context and problem description

2. Metaheuristics : GRASP

3. Column Generation / Branch and Bound

4 Results

5. Conclusions and perspectives

GRASP Framework

GRASP -Greedy Randomized Adaptative Search- (Feo & Resende, 1989)

- **GRASP** combines construction method (greedy heuristics), randomization and local search
- **GRASP** is a method which can be divided in two steps: a construction of an initial solution and improvement using local search. These two steps are repeated several times.
- **Principle for one iteration:**
  - Construction phase
    1. All the unrouted nodes are valuated with the greedy function.
    2. The n (parameter of the GRASP) best nodes following the greedy function are selected.
    3. Among these selected nodes, the node which will be inserted is randomly selected.
    4. If all the nodes are routed: end construction phase
       Else start to 1.
  - Local Search.
- We studied two GRASP versions: classical GRASP and Hybrid GRASP

Classical Grasp

1. Construction step : Best insertion

Choices:

- **where ?** → in which route? between which nodes?
- **how much ?** → quantity for the delivery?
- **at what time ?** → time of visit in the route?

2. Local search for « just in time version »:

- We chose the combination named: $E_o D_o O^+$.
  - E_o: "String Exchange"
  - D_o: "String Relocation"
  - O: "2-Opt"

2. Local search for version with inventory:

- We chose the combination named: $E_i E_o D_o D_i O A^+$.
  - E_i: "2-Exchange"
  - E_o: "String Exchange"
  - D_i: "Or-opt"
  - D_o: "String Relocation"
  - O: "2-Opt"
  - A: "Day-Exchange"

Hybrid Grasp

1. Construction step : Best insertion

2. Local search: using LNS –Large Neighbourhood Search (Shaw, 1998)

- Explores the neighbourhood of the solution by selecting a number of visits to be removed from the routing plan and re-inserting later these visits.
- How to choose the removed nodes:
  - Randomly
  - Best contribution: the nodes which cost the most
  - Best Resemblance: the nodes which have the biggest resemblance
- To find the reallocation we use **constraint programming techniques** with the solver choco.
- Branching technique: **Limited Discrepancy Search – LDS** (Harvey and Ginsberg ,1995)
  Parameter: the number of decision points at which we do not follow the heuristic (discrepancies).
- Introduction of a **tabu list** in the LNS technique.

Plan

1. Context and problem description

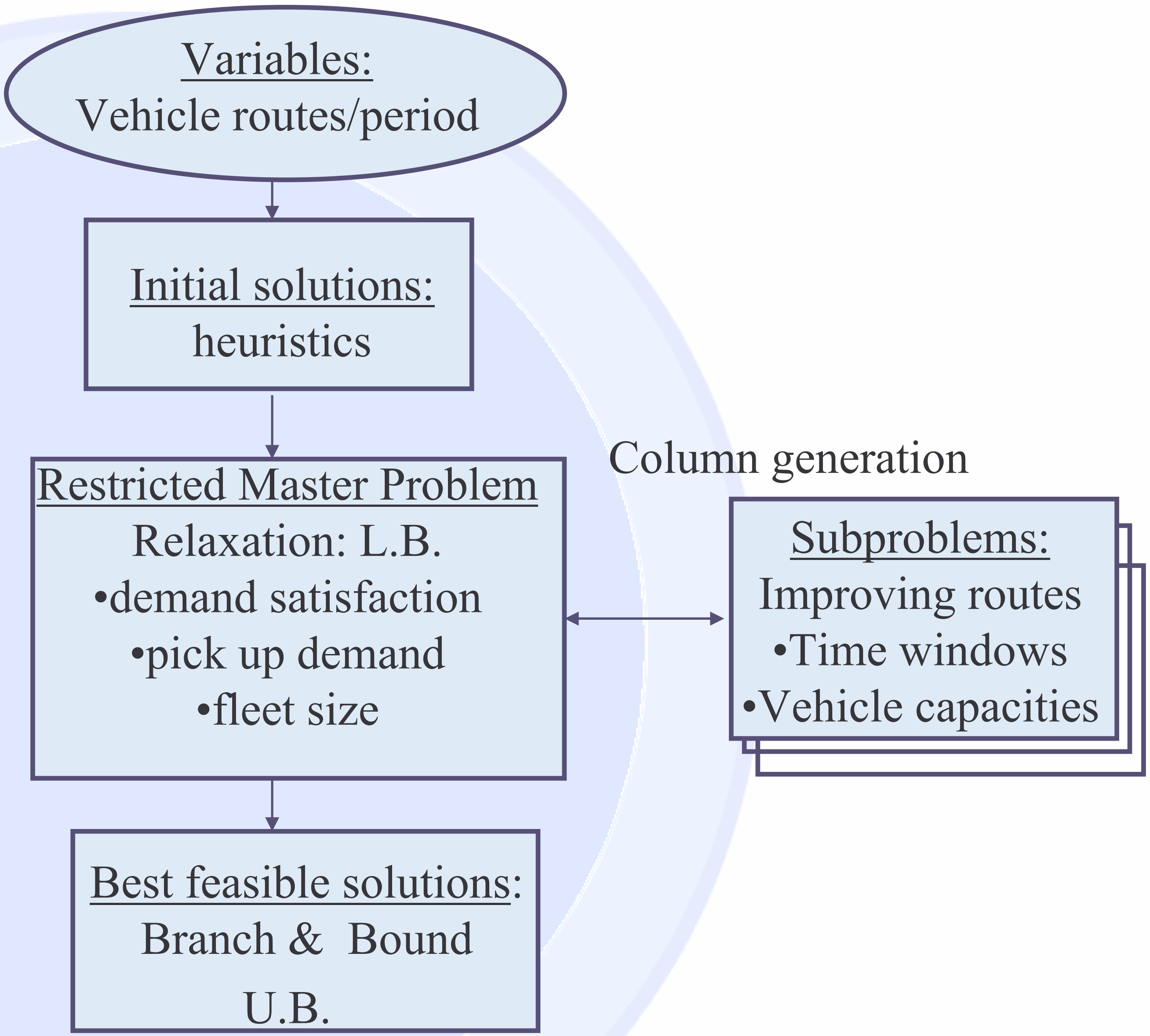
2. Metaheuristics

3. Column Generation / Branch and Bound

4 Results

5. Conclusions and perspectives

Set partitionning solution mechanism



Subproblem solution techniques

- Dynamic programming: ESPRC (Feillet et al, 2004)
  - -ressources : cost, load, time of visit

- Tabu search:
  - -neighborhoods: add, withdraw of site, exchanges, site moves
  - - finish with dynamic programming

- Constraint programming (Rousseau et al, 2002)
  - -variables: predecessors, successors, time of visit, vehicule load
  - -Constraints: TW, vehicule load, sub cycles forbidden

- Constraint programming with global constraint (Rousseau et al, 2002)
  - -variables: predecessors, successors, time of visit
  - - Constraints: same + cumulative constaint for vehicule load (Aggoun & Beliceanu, 1993)

Plan

1. Context and problem description

2. Metaheuristics

3. Column Generation / Branch and Bound

4 Results

5. Conclusions and perspectives

Instances

- Based on Solomon's instances for VRPTW
- 1848 instances of 25 sites for metaheuristics
- 108 instances of 5 sites (« exact method »)
- 96 instances of 6 sites (« exact method »)

1848 Instances of 25 sites

- Based on Solomon instances for the VRPTW: 25 sites.
- Inventory management: generation of the missing values.
- Size of the sites: small, average and big.
- Generation of 11 categories of instances with different repartitions of store sizes.
- 168 samples in each categorie – 1848 samples sum-total

Name	% small	% average	% big
A	100	0	0
B	0	100	0
C	0	0	100
D	34	33	33
E	5	25	70
F	25	25	50
G	50	25	25
H	70	25	5
I	25	50	25
J	25	70	5
K	5	70	25

- Solomon Instances: R1, R2, C1, C2, RC1, RC2 (56 instances).
  - 56 instances have fix demand depending on the size of the site.
  - 56 instances have Gaussian distribution demand: average = fix demand and deviation = 5.
  - 56 instances have Gaussian distribution demand: average = fix demand and deviation = 15.
- In total 56×3 instances for each categorie.

Instances characteristics

	Petit site	Grand site	Très Grand site
Capacité Stockage en nb palettes pleines	50	100	150
Demande Journalière en nb produits	20	40	80
	loi normale : $\mu = 20 \mid \sigma = 5$	loi normale : $\mu = 40 \mid \sigma = 5$	loi normale : $\mu = 80 \mid \sigma = 5$
	loi normale : $\mu = 20 \mid \sigma = 15$	loi normale : $\mu = 40 \mid \sigma = 15$	loi normale : $\mu = 80 \mid \sigma = 15$
Taux de retour journalier en nb produits	2	4	8
Stock initial palettes vides	5	10	20
Stock initial produit retour	2	4	8

Results : Classical GRASP

•Just in time:
- -improvement of heuristic methods by 3,5%
  - [2,9 to 4,3% depending of instance category]
- -Better results than heuristics on 83% of instances

• policy with storage:
- - improvement of heuristic methods by 4,5%
  - [3,4 to 5% depending of instance category]
  - -Better results than heuristics on 90% of instances

• no significant differences between categories of insta

• just in time vs storage :
  depends on ratio on storage vs transport cost !

•Split delivery : depends on fixed costs of tours!

Classical GRASP

- **Just in time:** Comparison of the results obtained with different size of candidates list with classical GRASP

Size: 3				Size: 5			
Average cost	Average time	≠ cost vs heuristics	≠ time vs heuristics	Average cost	Average time	≠ cost vs heuristics	≠ time vs heuristics
2 814 650	259 s	-3,5 %	× 142	2 842 621	257 s	-2,5 %	× 141

- **With inventory:** Comparison of the results obtained with different size of candidates list with classical GRASP

Size: 3				Size: 5			
Average cost	Average time	≠ cost vs heuristics	≠ time vs heuristics	Average cost	Average time	≠ cost vs heuristics	≠ time vs heuristics
2 714 059	258 s	- 4,5 %	× 139	2 736 858	257 s	-3,7%	× 139

Results : Hybrid GRASP

- works only on instances with small TW (363 instances)

- just in time: worst results than Classical GRASP by 5,5% on average
  - version with storage: worst results than Classical GRASP by10,5% on average

- Classical GRASP usually, but not always better than Hybrid GRASP

Hybrid GRASP

- Only on instances with small time windows (363 instances).
- Just in Time

Candidates List size	time for each iteration LNS	LDS parameter	Variable choice	≠ Classical GRASP (cost)
5	< 3 min || 70 ×	3	minDomain [1]	+6 %
3	< 3 min || 70 ×	3	minDomain	+5,5 %

- With inventory

Candidates List size	time for each iteration LNS	LDS parameter	Variable choice	≠ Classical GRASP (cost)
3	< 3 min || 100 ×	3	minDomain	+11,7 %
3	< 5 min|| 100 ×	3	minDomain	+10,7 %
3	< 5 min || 100 ×	5	minDomain	+10,5 %
3	< 5 min || 100 ×	1	minDomain	+12,2 %
3	< 5 min || 100 ×	2	DomOverDeg [2]	+ 13 %

[1] A heuristic selecting the variables with smallest domain

[2] A heuristic selecting the variables with smallest ratio (domainSize / degree) (the degree of a variable is the number of constraints linked to it)

Repartition of best solutions

	Just in time	storage
Classical heuristics	1%	0
Classical GRASP	91%	99%
Hybrid GRASP	8%	1%

Split delivery option

Pour l'option de partage

Version	Const. et Amélio. Nb min avec partage	Const. et Amélio. Nb min sans partage	GRASP Nb min avec partage	GRASP Nb min sans partage
CC = 2000 juste à temps	34 %	83 %	36%	64 %
CC = 20000 juste à temps	56 %	75 %	39 %	61 %
CC = 200000 juste à temps	61 %	69 %	47 %	53 %
CC = 2000 avec stockage	59 %	74 %	76 %	24 %
CC = 20000 avec stockage	59 %	71 %	32 %	68 %
CC = 200000 avec stockage	64 %	66 %	52 %	48 %

Comparaison des différentes versions selon les coûts de création d'une tournée

Results: Column generation + B.&B.

•Just in time:

-B.&B always yields an integer solution with small gap

-6 sites vs. 5 sites instances: computing time multiplied by 3 to 15 depending on sub problem method

-Best sub problem solution method :dynamic Programming (number of solutions within 1 hour)

-But 7,5% of 5 sites instances find a solution only with CP within 1 hour

• Policy with storage:

-B&B not relevant : integer solution found only in 22% of cases where relaxed solution obtained

- 6 sites vs. 5 sites instances: computing time multiplied by 2 to 19 depending on SP method

- Best sub problem solution method :dynamic Programming (number of solutions within 1 hour)

- But 90% of instances solved with the CP techniques

Just in time: Comparison of techniques

Les différentes techniques et les avantages des méthodes hybrides

Basée sur les méthodes exactes

Résultats obtenus en juste à temps (processeur Pentium IV à 2,8 GHz)

	5 sites				6 sites			
	ProgDyn	Tabou	Ppc	PpcCumu	ProgDyn	Tabou	Ppc	PpcCumu
Nombre problèmes résolus	93/108 **86 %**	100/108 93%	70/108 65 %	49/108 45%	71/96 **74 %**	61/96 64 %	34/96 35 %	28/96 29%
Temps moyen obtention solution relaxée	**30 s**	14 s	727 s	144 s	**475 s**	61 s	60 s	343 s
Nombre solutions entière obtenue	93/108 86%	100/108 93%	70/108 65 %	49/108 45%	71/96 74%	61/96 64%	34/96 35%	28/96 30%
Temps moyen obtention solution entière	8 s	0,1 s	1,7 s	35 s	244 s	24 s	8 s	185 s
Nombre colonnes	768	547	163	289	924	728	196	332
Nombre sol entières = sol relaxées	79/93 **85 %**	55/100 55 %	31/70 44 %	21/49 43%	54/71 **76 %**	29/61 48 %	9/34 26 %	8/28 29 %
≠ sol entières sol relaxées	**0,3 %**	1,3 %	5%	4 %	**1,7%**	3 %	11 %	4,5%

Plan

1. Context and problem description

2. Metaheuristics

3. Column Generation / Branch and Bound

4 Results

5. Conclusions and perspectives

Conclusions

✓Problem:

•We have proposed a « realistic »framework for multiperiodic routing with reverse flows and a variety of management policies

• Tests made on a large number of random data sets with different characteristics

✓Techniques:

• GRASP method improves classical heuristics for 25 sites instances (with more computing time)

• Column Generation + B.&B. find solutions close to optimum for small instances for Just in Time policy

• Classical methods for Grasp and column generation usually better than hybrid methods, but not always

Perspectives

✓Problem:

- Possible extensions
- Tests on other random instances : Cordeau et al. (1997)
- Tests on real application

✓Techniques

- classical heuristics: automatically generate best combination
- develop other metaheuristics than GRASP
- develop more efficient CP techniques : global constraint for GRASP
- « exact methods » : develop Branc and Price Technique for policy with storage
- Implement valid inequalities…