Stability in waste collection problem : a case study

Frédérique BANIEL (University of Toulouse : LGP / ENIT - MOGISA / LAAS-CNRS)
Marie-José HUGUET (University of Toulouse : MOGISA / LAAS-CNRS)
Thierry VIDAL (University of Toulouse : LGP/ ENIT)

Odysseus 26-29 May 2009

Plan

1. Problem under study

2. Modeling

3. Algorithms proposed

4. Experimentations

5. Conclusion et Open issues

Context

- **Goal : Maintain stability in the waste collection routes when**
    - **Reorganization** of routes is necessary in a local authority
  - 4 main objectives :
    - Collection costs
    - Service quality for users
    - Environnemental impact
    - Employees works conditions
- **Partnership :**
  - Local authority : **CAM** problem and datas

Context

- **Organisation of the local authority :**
  - Single depot and disposal site.

- **High Constraints :**
  - All houses must be collected,
  - Vehicle number,
  - Maximal capacity for each vehicle,
  - Time windows,
  - Work time for employees is bounded.

- **Modeling : Vehicle Routing Problem with Time Windows or VRPTW**

Vehicle Routing Problem with Time Windows

Minimize :

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} \sum_{k=1}^{m} x_{ij}^{k} \quad (eq1)$$

Constraints :

$$\sum_{i=1}^{n} x_{ij}^{hk} = y_j^{hk} \; (j = 2,\ldots n; k \in K; h \in H)$$

$$x_{ij}^{hk} \in \{0,1\}$$

$$\sum_{j=2}^{n} x_{ij}^{hk} = y_i^{hk} \; (i = 2,\ldots n; k \in K; h \in H)$$

$$\sum_{i \in Y'} t_i^{hk} . y_i^{hk} + \sum_{i \in Y'} \sum_{j \in Y'} x_{ij}^{hk} . t_{ij}^{k} \le T \; (k \in K; h \in H)$$

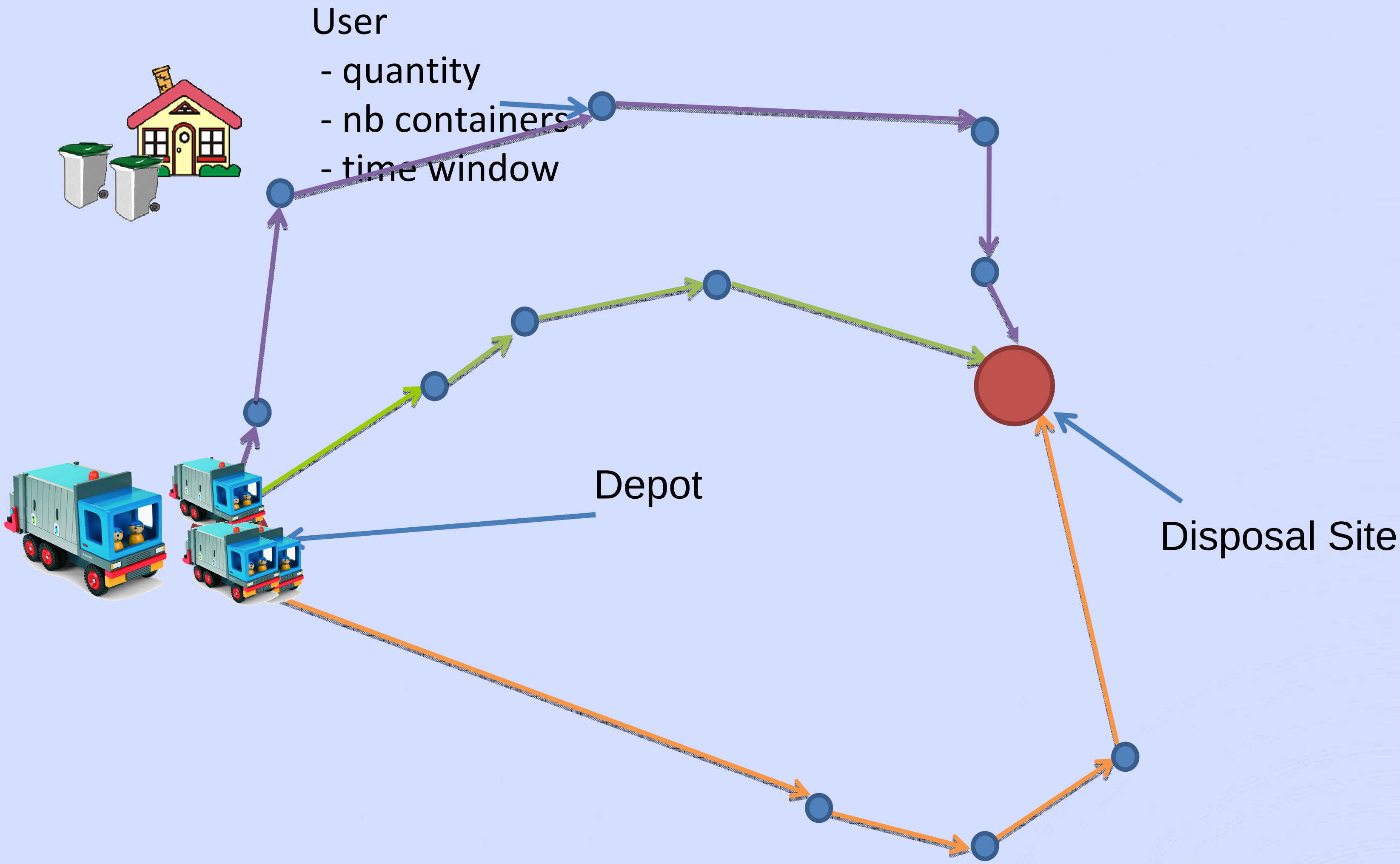
$$\sum_{i \in Y'} q_i^{hk} . y_i^{hk} \le Q^k \; (k \in K; h \in H)$$

$$\sum_{(i,j) \in s^2} x_{ij}^{hk} \le |S| - 1 (S \subset Y'; 2 \le |S| \le n-1; k \in K; h \in H)$$

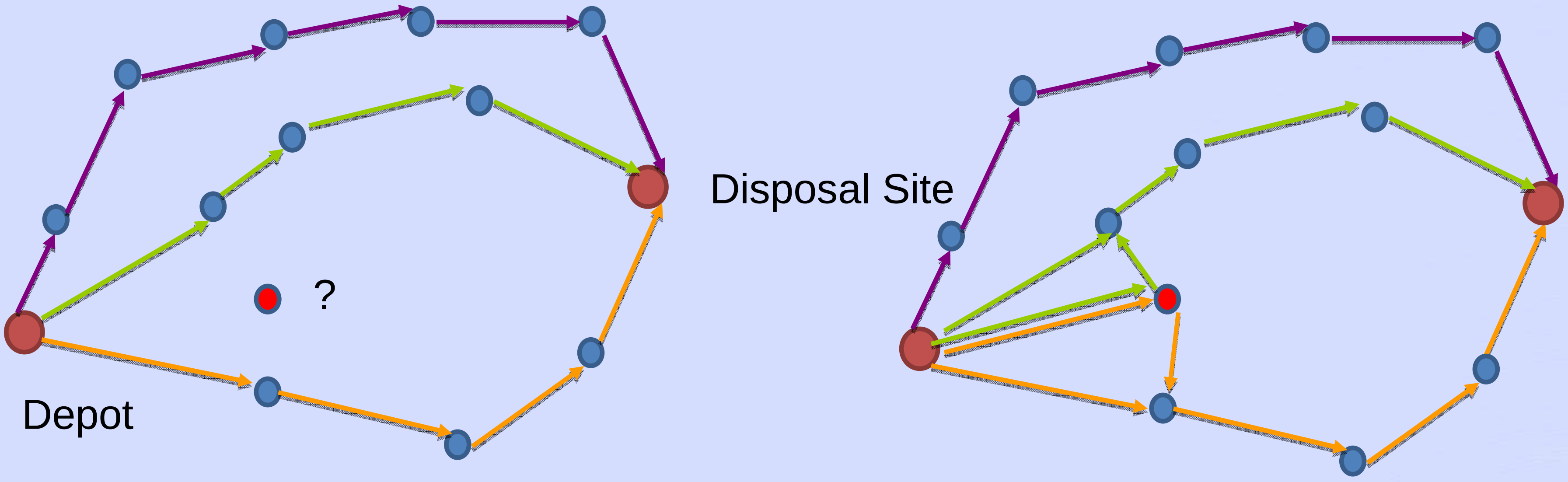
$$y_i^{hk} \times est_i^h \le y_i^{hk} . st_i^{hk} \le y_i^{hk} \times \left(lft_i^h - t_i^{hk} \right)$$

$$st_i^{hk} + t_i^{hk} + t_{ij}^{k} - M \times \left(1 - x_{ij}^{hk} \right) \le st_j^{hk}$$

Waste collection problem



Presentation of dynamic problem



- **Reorganization :**
  - Population increase about **2%** per year,
  - Waste quantity increase about **2%** per year and per inhabitant,
  - Methods proposed to update routes

Criteria for the reorganization

GOAL : obtain a solution with a low cost and which disturb as low as the initial solution

- **Collection costs (= distance)**
- **Stability**
  - **In general :**
    - **Gap** in distance and in work time.
  - **Service quality (= users point of view) :**
    - modification of collection hour ➔ **hour gap**.
  - **Work conditions (= employees point of view) :**
    - Adding/deleting collection points in collection routes,
    - Collection order moved.

    ➔ Composition Gap and order gap

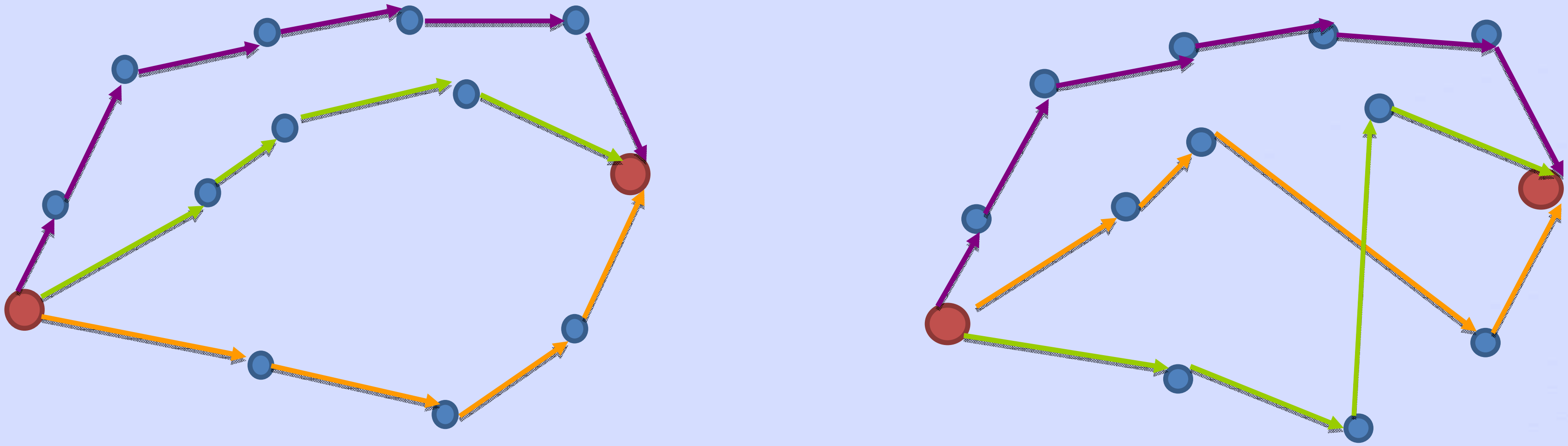
Methods used

- **Problem 1 :** Insertion of new nodes
  - I1 : Insert in the nearest neighborhood in respecting constraints,
  - I2 : I1 + local search (string -exchange) between routes affected by insertion and its close routes,
  - I3 : I1 + local search (string–exchange + or-opt) between all pairs of routes
  - I4 : I1 + tabu search,
  - I5 : I1 + bi-objective tabu search,
  - I6 : building from scratch.

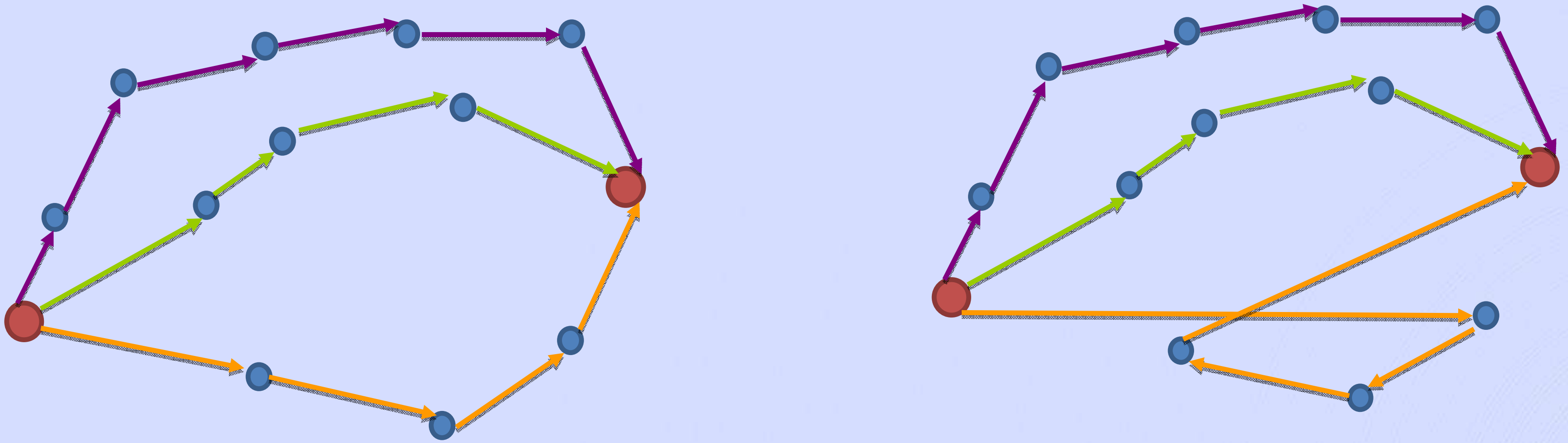
- **Problem 2 :** Quantity increase
  - I1 : Find an admissible solution,
  - I2 : I1 + local search (string–exchange + or-opt) between all pairs of routes
  - I3 : I1 + tabu search,
  - I4 : I1 + bi-objective tabu search,
  - I5 : building from scratch.

The Local Search Algorithm

- String - Exchange Algorithm



- Or-Opt Algorithm

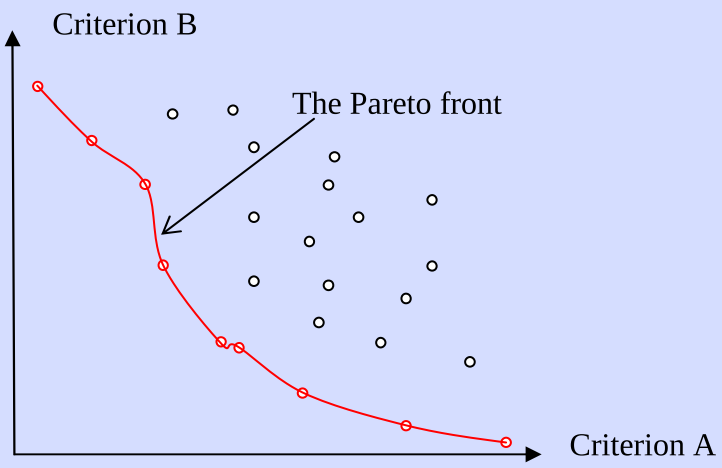


The Tabu Search

- Mono or Multi-objective algorithm
- Criteria used : the distance and stability
- Steps of the algorithm :

1. From an initial solution → neighbourhood
2. Deletion of the solutions which are in both the Tabu list and the neigbourhood
3. Creation of the Pareto front from the new neighbourhood
4. Random selection in the Pareto front → new initial solution
5. Add new initial solution to the Tabu list

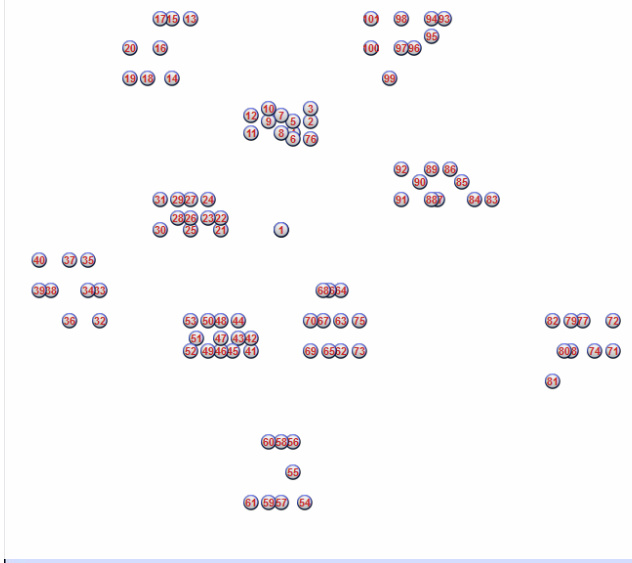
The Multi-Creteria Tabu Search



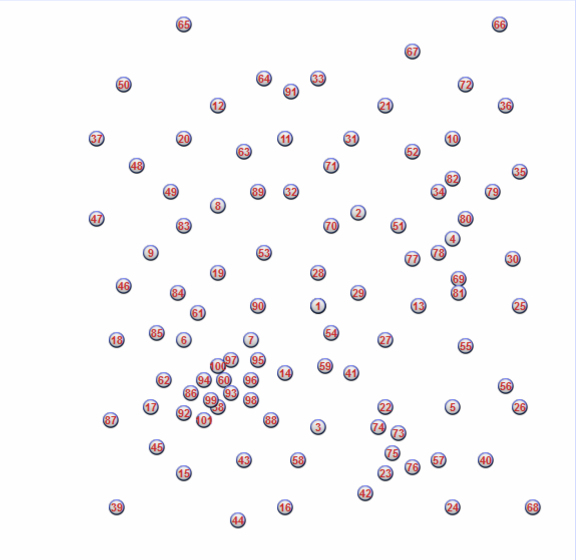
- Keep only one solution :
  - Compromise between distance and stability.

Experimentations

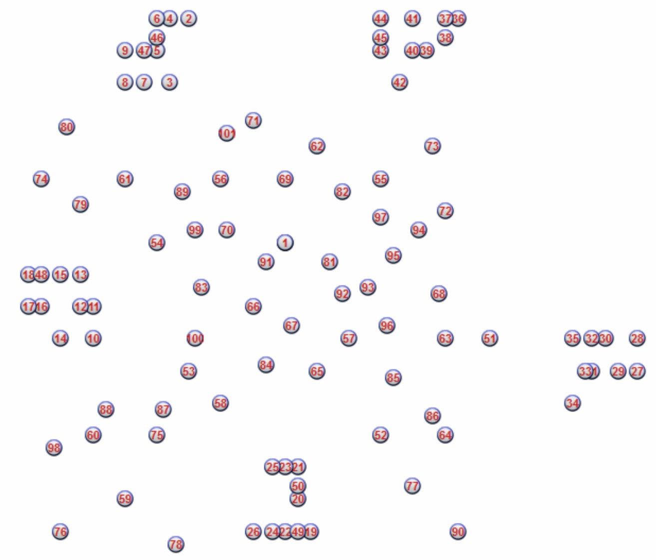
- Instances used :
  - Solomon’s instances : 3 types C, R and RC,
  - 100 nodes to collect,
  - VRP Time Window.



Instance C



Instance R



Instance RC

- Modifications :
  - Modifications are needed in order to treat considered problems.

Experimentations

- Insertion new nodes
  - Modifications :
    - Delete nodes at random ➔ Initial solution
    - Insert nodes deleting➔ New solution

			Stability				
			General		Users	Employee	
Instance	Method	Distance	Distance Gap	Time Gap	Hour Gap	Composition Gap	Order Gap
C1	1	1052	166	512	906	9	8
	2	1015	146	510	1654	12	10
	3	997	149	504	1594	13	10
	4	985	131	695	3136	13	21
	5	1002	146	663	2641	20	21
	6	864	116	689	5911	13	31

Experimentations

- Quantity improving
  - Modifications :
    - Decrease of quantity in 10 nodes about 20% ➔ Initial Solution
    - Increase their quantity ➔ New Solution

			Stability				
			General		Users	Employee	
Instance	Method	Distance	Distance Gap	Time Gap	Hour Gap	Composition Gap	Order Gap
C1	1	916	133	110	66	1	0
	2	869	60	360	4880	1	26
	3	866	84	426	2648	12	19
	4	889	41	217	2048	2	14
	5	864	61	325	5217	0	26

Conclusion and open issues

- We propose methods :
  - To insert new nodes,
  - To update quantity in nodes.

- Results :
  - Heuristics methods
  - Metaheuristic methods
  - Multi-criteria methods

- Perspectives :
  - Develop multi-criteria methods in choosing different compromize between criteria.