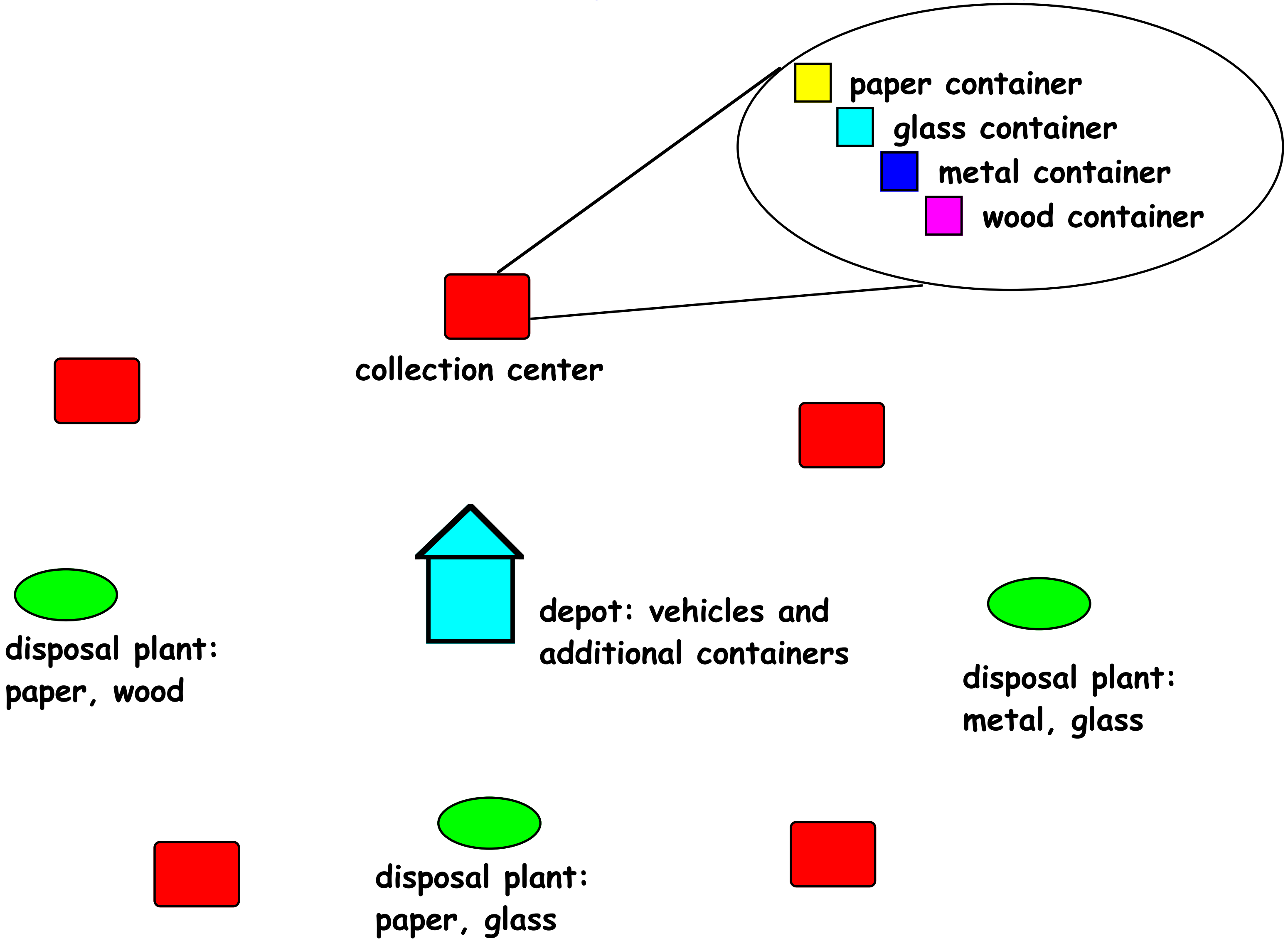
A particular Vehicle Routing Problem arising in the collection and disposal of special waste

Roberto Aringhieri
Maurizio Bruglieri
Federico Malucelli
Maddalena Nonato

http://www.elet.polimi.it/upload/malucell

TRISTAN V - 2004

Problem description: service type



Problem description: operations

- Users convey waste to their nearest collection center and dispose it into the appropriate container
- Once a container is full the collection center issues a **service request** consisting in emptying the full container
- The company operates a **swap** between a full container and an empty one, disposing the waste in the nearest disposal center
- The swap takes place when the collection center is closed: the removal and substitution of a container may take place in different moments and not necessarily in this order

Problem description: optimization aspects

- A vehicle can carry one container at a time
- The containers are owned by the company
  ⇒ containers are not obliged to return to the original center
- A container, once emptied, can be reused for other materials
  ⇒ compatibility constraints
- Several types of containers (left, right, with compactor…)
  ⇒ compatibility constraints
- Limited number of spare containers at the depot
- Maximum duration of a vehicle route
  ⇒ minimize vehicle number and the total traveled time

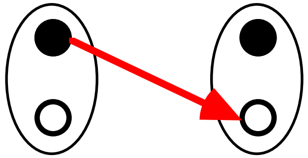
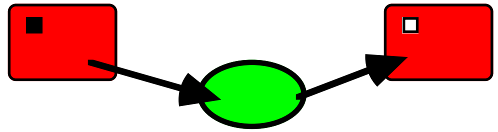
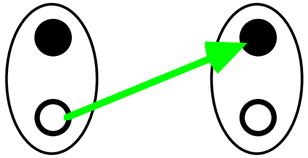
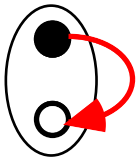
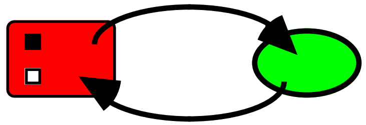
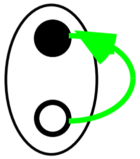
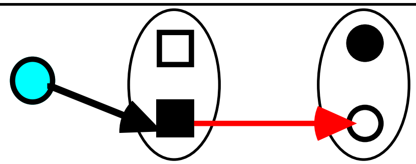
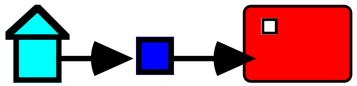
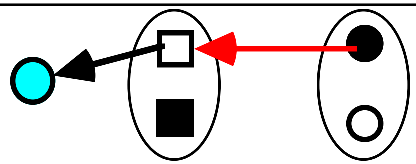
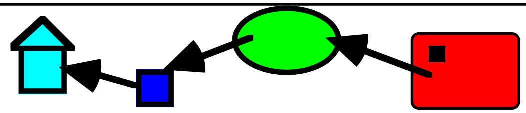
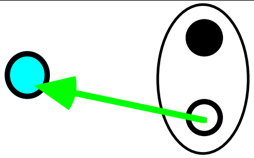
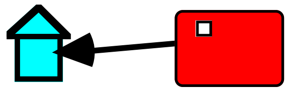
Problem description: containers

Vehicle Routing graph construction

Nodes

Physical graph	Vehicle Routing Graph
service request	● full container ○ empty container
depot	depot node
spare container	□ dummy "empty" node ■ dummy "full" node
disposal plant	hidden in the arcs

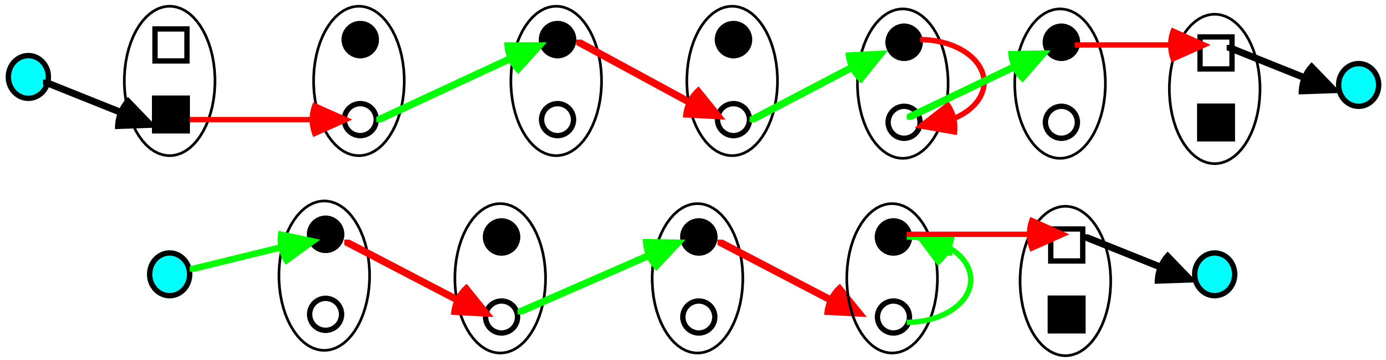
Vehicle Routing graph construction: some arcs

VR graph	Physical graph	vehicle	"cost" of the arc
		loaded	from centers *i* and *j* passing by the closest dump
		unloaded	from centers *i* and *j*
		loaded	from center *i* the closest dump and back to *i*
	container swap	no travel (unloaded)	only loading unloading times
		loaded	from depot to center *i*
		loaded	from center *i* to depot passing by the closest dump
		unloaded	from center *i* to depot
...	...	...	...

Vehicle routing: routes

Loaded arcs join compatible nodes (i.e., same type of container)

Route: close path on the depot



Alternating sequence of loaded and unloaded arcs
(full and empty containers)

Solution: set of routes covering all (round) nodes

Objective: minimize the total traveled time and the number of vehicles (i.e., arcs leaving the depot)

Related work

[1] C. Archetti, M.G. Speranza
Collection of waste with single load trucks: a real case
www.eco.unibs.it/dmq/speranza
no container circulation

[2] L. Bodin, A. Mingozzi, R. Baldacci, M. Ball
The Rollon-Rolloff Vehicle Routing Problem
Transportation Science 34 (3) 271-288 (2000)
disposal plant in the depot

[3] L. De Muelemeester, G.Laporte, F.V.Louveaux, F. Semet
Optimal Sequencing of Skip Collections and Deliveries
Journal of Operational Research Society 48, 57-64 (1997)
unbounded number of spare containers

Asymmetric VRP

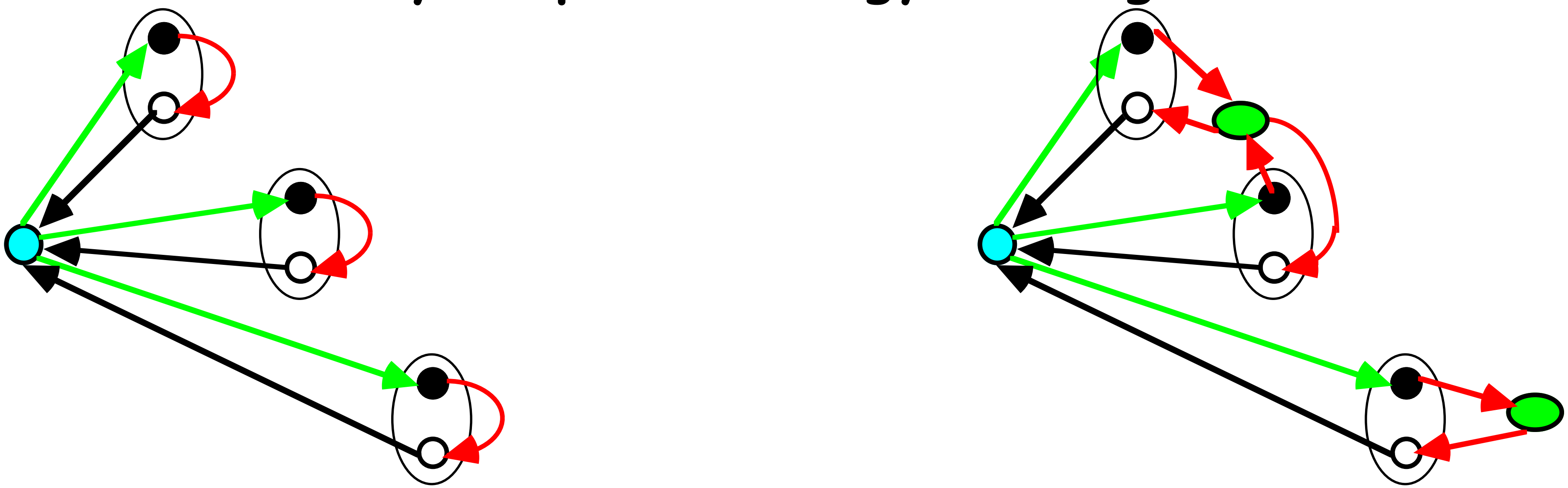
- Asymmetric travel times
- Alternating arcs
- Almost bipartite graph (bipartite if we split the depot node)
- Compatibility constraints (sparsification of the graph)
- Route duration constraints

Mathematical model

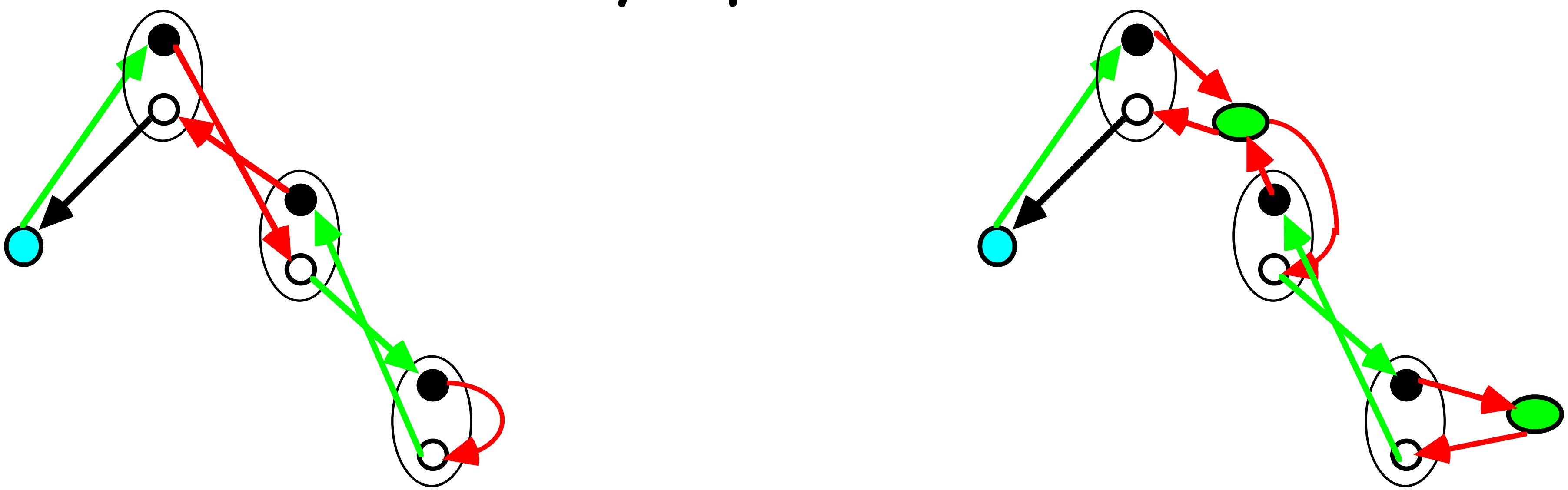
Commercial MP software fails to solve instances with a dozen of requests

Company solution

Most usual currently adopted strategy: "triangles"



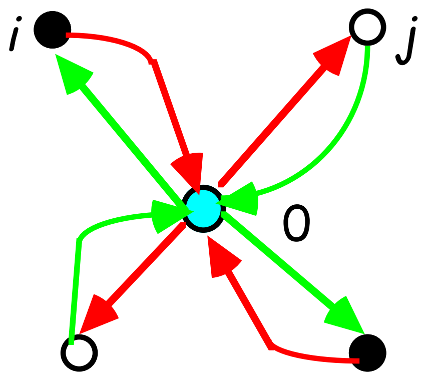
The solution can be trivially improved...



Constructing a feasible solution

Modified Clarke and Wright

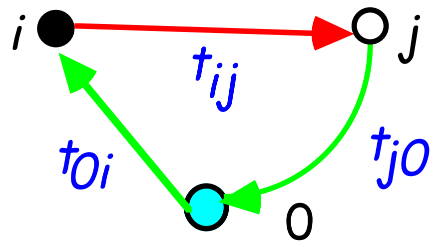
Starting configuration:



Note that the solution can be infeasible w.r.t. the available spare containers

1) Savings computation:

for each pair (*i*,*j*) of compatible nodes: $s_{ij} = t_{ij} - t_{i0} - t_{0j}$



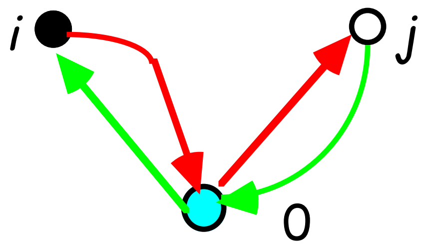
2) Sort the savings in non increasing order

3) Greedy phases:

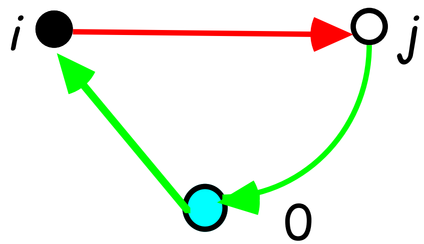
Phase 1

- consider the savings in the order
- make the shortcuts that decrease the infeasibilities

  (i.e., decrease the use of spare containers)



1 spare container



0 spare containers

Phase 2

- consider the other savings in the order
- make the other shortcuts

All shortcuts are performed only if the resulting route has length not exceeding the maximum

Lower bounds on the total travel time

Match the savings in the best possible way [3]

$$\max \sum_{i,j} s_{ij}\, x_{ij}$$

$$\sum_{i} x_{ij} \le 1 \quad \forall\, j$$

$$\sum_{j} x_{ij} \le 1 \quad \forall\, i$$

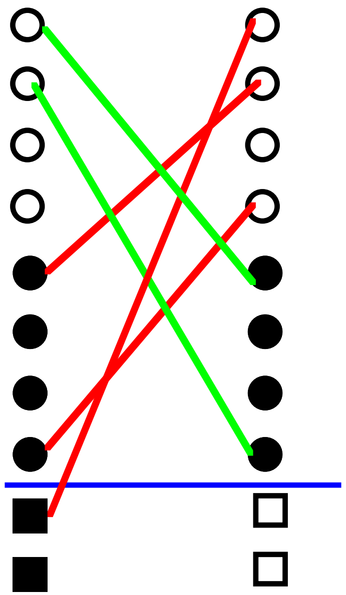
$$x_{ij} \ge 0 \quad \forall\, i,j$$

Minimum total time cycle cover of the graph

Refinement of the lower bound

Extension to the case with a bounded number of spare containers

Include also the dummy nodes corresponding to spare containers in the cycle cover matching problem



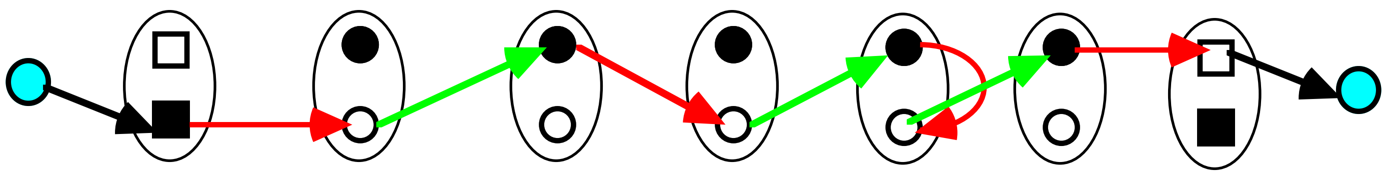
LP based bound

- Consider the AVRP formulation
- Relax the integrality on arc variables
- Keep integrality on variable counting the number of vehicles z
- The bound is computed by performing a binary search on z
  At each iteration solve an LP

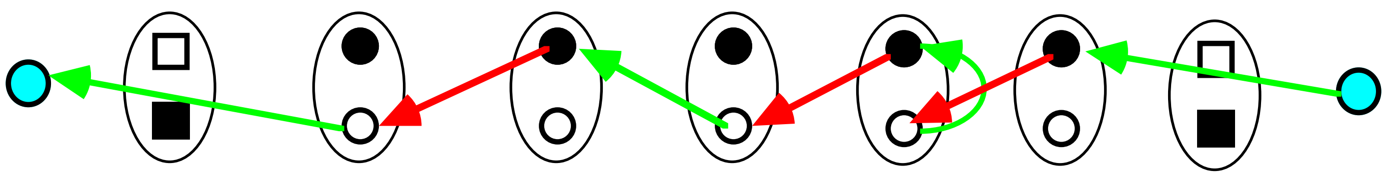
Improving the solution: Local Search

12 different types of neighborhoods considering:

- inter route, intra route
- alternating loaded-unloaded arcs
- spare containers use
- reversing routes (or portion of routes) to save containers



original route

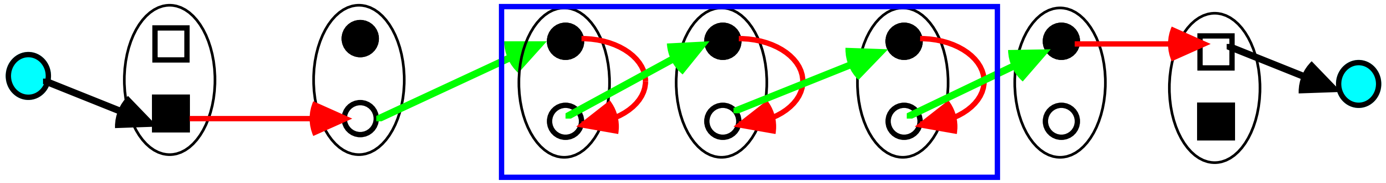


reversed route saving a container

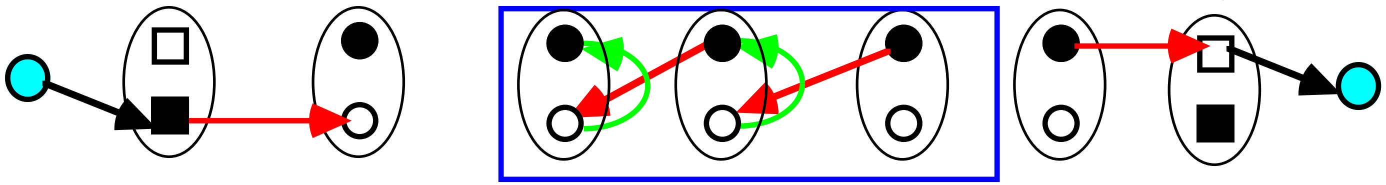
Reversing co-sited loaded arcs

Loaded arcs are very time consuming

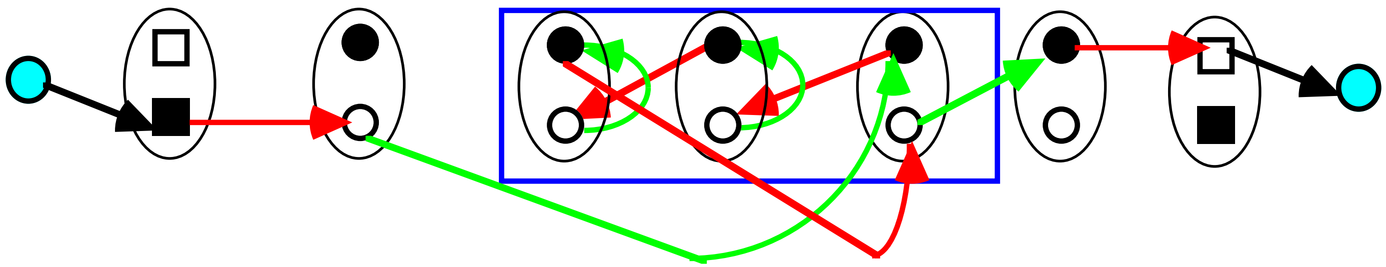
Unloaded arcs inside the same center are very "short"



Reversing a sequence of a co-sited loaded arcs my be interesting



remove the sequence and reverse it



insert the new sequence in the previous solution

Local Search control algorithm

while the solution improves **do**
  for $i=1,\ldots,12$ **do**
    Local Search with neighborhood N_i

The Local Search performs the exhaustive search inside the neighborhood and selects the best improvement

Real case

Regional area in central Italy of about 4000 Km^2

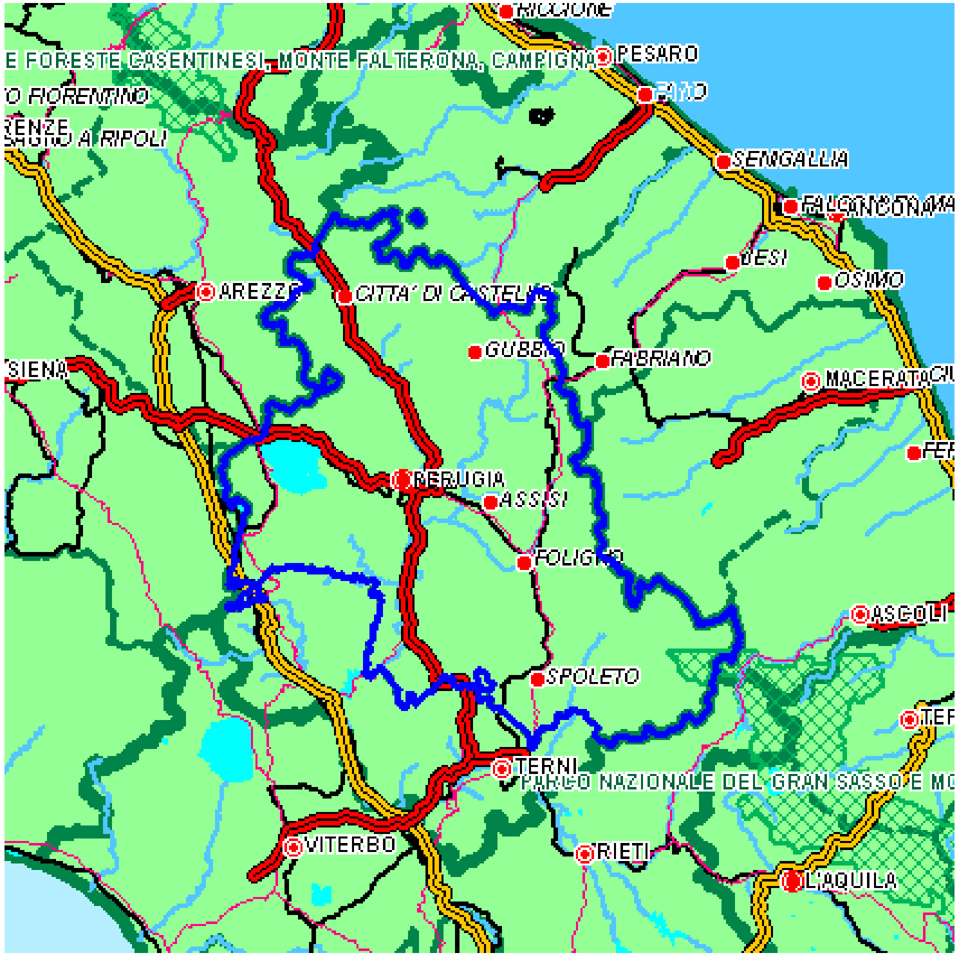
10 collection centers

6 types of containers

10 types of material

3 disposal plants

Max route duration 375 min.



Results on real cases

Day	requests	company solution	CPLEX	Cicle cover bound	LP based bound	MCW	LS	CPU Cplex	CPU LS
17/11	8	789	**668**	620	650	668	668	3.52	0.04
18/11	3	325	**325**	230	262	325	325	0.1	0.02
19/11	7	615	**615**	573	601	615	615	3.44	0.03
20/11	8	813	**701**	657	685	709	709	7.56	0.03
21/11	6	686	**665**	547	594	665	665	0.1	2.48
22/11	9	1001	**843**	801	843	903	903	21.32	0.02
24/11	8	712	**684**	642	670	698	698	3.05	0.06
25/11	8	672	**575**	537	551	608	586	19.42	2.67
26/11	6	679	**599**	554	583	606	606	6.71	0.04
27/11	8	975	**839**	727	772	839	839	5.78	0.02
28/11	6	699	**606**	564	592	624	624	0.1	0.02
29/11	11	1075	**882**	840	882	948	948	0.1	0.06

LS times in seconds on a Pentium 2 GHz

CPLEX times in seconds on a biprocessor Xeon 2.8 GHz

Randomly generated instances

Real network

40 - 60 - 80 requests

Different numbers of available spare containers:

- **T0 none**
- **T1 one for each type**
- **T2 ∞**
- **T3 an intermediate number**

Preliminary and partial results

Requests/Type	Cycle cover bound	LP based bound	MCW	LS	CPU LS sec	Gap MCW %	Gap LS %
R40.T2	3549	3586	3723	3684	2.71	3.8	2.7
R40.T0	3510	3561	4195	4195	5.25	17.8	17.8
R40.T1	3510	3552	4034	3978	6.09	13.6	11.9
R40.T3	3510	3552	4040	3984	10.72	13.7	12.2
R40.Tdef	3510	3552	4027	3971	3.83	13.4	11.8
R40.T3	3335	3372	3688	3648	10.44	9.4	8.2
R40.T2	3475	3512	3632	3612	4.12	3.4	2.8

Conclusions and future work

- Modified CW gives good results
  - restart procedure (randomization)
- More sophisticated LS based procedures
  - Variable Neighborhood Search
- Investigate a more specific mathematical model
- Multidepot case
- Extension to industrial waste
- Multiperiod planning